VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH TECHNOLOGIÍ
FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION

ÚSTAV MIKROELEKTRONIKY
DEPARTMENT OF MICROELECTRONICS

SYSTÉM VYHODNOCOVÁNÍ PRO STOPOVÝ DETEKTOR V PEVNÉ FÁZI
MEASUREMENT SYSTEM FOR ETCHED TRACK DETECTOR

DIPLOMOVÁ PRÁCE
MASTER'S THESIS

AUTOR PRÁCE — **Bc. Juraj Galbavý**
AUTHOR

VEDOUCÍ PRÁCE — **doc. Ing. Radovan Novotný, Ph.D.**
SUPERVISOR

BRNO 2017

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH TECHNOLOGIÍ

Diplomová práce

magisterský navazující studijní obor **Mikroelektronika**
Ústav mikroelektroniky

Student: Bc. Juraj Galbavý ***ID:*** 119915
Ročník: 2 ***Akademický rok:*** 2016/17

NÁZEV TÉMATU:

Systém vyhodnocování pro stopový detektor v pevné fázi

POKYNY PRO VYPRACOVÁNÍ:

Stopový detektor v pevné fázi z plastového polymeru CR-39 lze použít pro registraci těžkých nabitých částic. Díky tomu lze vyhodnocovat závislost mezi počtem zviditelněných stop dopadu alfa záření na dozimetrickém skle a absorbovanou dávkou záření. Ke kvantifikaci je nutné vyhodnotit snímky zachycené mikroskopem. Vymezte algoritmus umožňující adekvátní opakovatelné stanovení počtu stop, bez započtení nečistot na skle, vlivu poškrábání a jiných neshod na snímcích. Navrhněte vhodnou metodu detekce stop na snímku. Výsledky porovnejte s historickými daty. Výstupní informací navrženého systému vyhodnocení bude protokol shrnující odhad počtu stop v dílčích segmentech a celkový počet stop. Finálně navrhněte aplikaci umožňující automatické zpracování dat z dozimetru. Zohledněna bude potřeba automatického zpracování a minimalizace falešné detekce.

DOPORUČENÁ LITERATURA:

Podle pokynů vedoucího práce.

Termín zadání: 6.2.2017 ***Termín odevzdání:*** 25.5.2017

Vedoucí práce: doc. Ing. Radovan Novotný, Ph.D.
Konzultant:

doc. Ing. Lukáš Fujcik, Ph.D.
předseda oborové rady

ABSTRAKT

Cieľom tejto práce je navrhnúť algoritmus pre automatické spočítanie stôp na snímkach z detektoru stôp v pevnej fázi vyrobeného z polyméru CR-39. Stopy sú produkované alfa časticami. Chemicky vyleptaný detektor je snímkovaný pod mikroskopom a po vytvorení 64 snímok segmentov na povrchu detektora je nutné automaticky rozpoznať stopy kruhového tvaru v obraze. Práca sa zameriava na otestovanie kruhovej Houghovej transformácie na detegovanie stôp. Výsledný softvér by mal automatizovať počítanie stôp na jednom detektore a mal by byť odolný voči nekvalitnému obrazu a voči znečisteniu na povrchu detektoru. Softvér vyprodukuje výstupný protokol merania s celkovým počtom stôp a počtom stôp na jednotlivých segmentoch.

KĽÚČOVÉ SLOVÁ

detektor stôp v pevnej fázi, CR-39, dozimetria, kruhová Houghova transformácia, automatické počítanie objektov, systém merania

ABSTRACT

The aim of this thesis is to design an algorithm for an automatic track counting of an image of etched track detector made of CR-39 polymer. Tracks are produced by alpha particles. Chemically etched detector is imaged using a microscope resulting in 64 images of segments on the surface of the detector. Circle shaped tracks in the images have to be detected and counted. This thesis evaluates the utilization of circle hough transform for circle detection. The final software should automate a detector track counting and should also account for defects in the image and contamination of detector surface. The software will produce a measurement report with a total track count in each segment.

KEYWORDS

etched track detector, CR-39, dosimetry, circle hough transform, automatic object counting, measurement system

Bibliografická citácia

GALBAVÝ, J. *Systém vyhodnocování pro stopový detektor v pevné fázi.* Brno: Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, 2017. 65 s. Vedoucí semestrální práce doc. Ing. Radovan Novotný, Ph.D.

SIX research centre
sensor, information and communication systems

Faculty of Electrical Engineering
and Communication

Brno University of Technology
Technicka 12, CZ-61600 Brno, Czechia

http://www.six.feec.vutbr.cz

Experimentální část této diplomové práce byla podpořena výzkumnou infrastrukturou vybudovanou v rámci projektu CZ.1.05/2.1.00/03.0072
Centrum senzorických, informačních a komunikačních systémů (SIX)
operačního programu Výzkum a vývoj pro inovace.

.

MINISTERSTVO ŠKOLSTVÍ, MLÁDEŽE A TĚLOVÝCHOVY

EVROPSKÁ UNIE
EVROPSKÝ FOND PRO REGIONÁLNÍ ROZVOJ
INVESTICE DO VAŠÍ BUDOUCNOSTI

2007–13
OP Výzkum a vývoj pro inovace

PREHLÁSENIE

Prehlasujem, že svoju diplomovú prácu na tému Systém vyhodnocování pro stopový detektor v pevné fázi som vypracoval samostatne pod vedením vedúceho diplomovej práce a s použitím odbornej literatúry a ďalších informačných zdrojov, ktoré sú všetky citované v práci a uvedené v zozname literatúry na konci práce.

Ako autor uvedenej diplomovej práce ďalej prehlasujem, že v súvislosti s vytvorením tejto diplomovej práce som neporušil autorské práva tretích osôb, najmä som nezasiahol nedovoleným spôsobom do cudzích autorských práv osobnostných a/alebo majetkových a som si úplne vedomý následkov porušenia ustanovení § 11 a nasledujúcich zákona č. 121/2000 Sb., o práve autorskom, o právach súvisejúcich s právom autorským a o zmene niektorých zákonov (autorský zákon), v znení neskôrších predpisov, vrátane možných trestnoprávnych dôsledkov vyplývajúcich z ustanovenia druhej časti, hlavy VI. díl 4 Trestního zákoníku č. 40/2009 Sb.

V Brne dňa 25. 5. 2017

.....

(podpis autora)

POĎAKOVANIE

Ďakujem vedúcemu diplomovej práce doc. Ing. Radovanovi Novotnému, Ph.D. za účinnú metodickú, pedagogickú a odbornú pomoc a ďalšie cenné rady pri spracovaní mojej diplomovej práce.

V Brne dňa 25. 5. 2017

.....

(podpis autora)

Obsah

Zoznam obrázkov

ÚVOD

V tejto práci je navrhovaný algoritmus pre spracovanie optických mikroskopických snímok dozimetrického skla z materiálu CR-39. [1][2][3] Dozimeter sa používa pre meranie koncentrácie radónu v obytných a pracovných priestoroch alebo pre kontrolu ožiarenia pracovníkov na pracoviskách s vyššou radiačnou záťažou, akými sú pracoviská v podzemných priestoroch s vyšším výskytom radónu, ako sú jaskyne alebo bane. [1] [8] [9]

V práci sa používa dozimeter z polymérového materiálu CR-39, ktorý má vysokú citlivosť na alfa žiarenie, ktoré je produkované radónom a jeho dcérskymi produktmi rozpadu. Jeden zásah alfa časticou sa na vyvolanom dozimetri prejaví ako mierne zakrivená stopa z viacerých bubliniek. Väčšina zásahov na povrchu pod mikroskopom je vidieť ako jedna veľká bublina. Tieto tmavšie okrúhle bubliny sa dajú spočítať a ich počet priamo úmerne odpovedá absorbovanej dávke žiarenia pracovníka alebo koncentrácii radónu vo vzduchu.

Dozimetre, ktorými sa zaoberá táto práca, sa spracúvajú na Slovensku na Oddelení radiačnej hygieny Slovenskej Zdravotníckej Univerzity v Bratislave. [10] Súčasný softvér spracúvajúci snímky síce automaticky narátava počet zaznamenaných častíc alfa žiarenia, avšak má problémy s falošnými detekciami spôsobenými škrabancami na snímkovanom skle.

Úlohou tejto práce je vyvinúť nový softvér, ktorý bude odolný voči poruchám v obraze a bude schopný zo snímok narátavať počet zachytených častíc s minimálnou potrebou zásahu laboranta, ktorý ho ovláda. Pre tento účel má byť nájdený účinný algoritmus, ktorý predspracuje obraz a nakoniec spočíta množstvo zásahov alfa časticou s minimálnym počtom nedetegovaných zásahov a s minimálnym počtom falošných detekcií. Softvér má používať navrhnutý algoritmus spracovania obrazu, skrátiť dobu spracovania jednej vzorky a má byť užívateľsky prívetivý.

1 TEORETICKÝ ROZBOR

V tejto kapitole je najprv vysvetlená potreba merať ožiarenie pracovníkov ionizujúcim žiarením, následuje popis meracej metódy a použitého detekčného materiálu. Popísaný je aj súčasný stav vyvíjaného meracieho systému.

1.1 Vplyv ožiarenia radónom na zdravie

Zdravotné riziká radónu sú spôsobené jeho krátko žijúcimi produktmi premeny. Radón je prírodný rádioaktívny plyn bez vône, chuti a zápachu. Má protónové číslo 86 a v prírode je zastúpený tromi rádionuklidmi, z ktorých najzávažnejší z hľadiska zdravotných účinkov je izotop Rn^{222}, ktorý vzniká rádioaktívnou premenou U^{238}. Dcérske produkty a aj radón samotný sa inhaláciou dostávajú do pľúc, kde sa vďaka krátkemu polčasu premeny rozpadajú a emitujú alfa žiarenie. Žiarenie poškodzuje tkanivo v pľúcach, čo môže vyvolať rakovinú pľúc. Rakovina sa neobjaví okamžite, príznaky sa prejavia po 10 až 30 rokoch od expozície. Dva krátko žijúce produkty dominujú v depozícii alfa žiarenia v pľúcach, a to Po^{218} a Po^{214}. [1]

Radón a jeho dcérske produkty rozpadu sú zodpovedné za približne polovicu celkovej dávky ožiarenia obyvateľstva. Radón vyviera z geologického podložia a rozptyľuje sa vo vzduchu alebo preniká do vnútorných priestorov budov, alebo sa koncentruje v jaskyniach. Koncentrácia radónu na otvorenom priestranstve je celosvetovo veľmi nízka, vyjadrená v objemovej aktivite je 40 $Bq.m^{-3}$. Oproti tomu v miestostiach budov je objemová aktivita rádovo v stovkách $Bq.m^{-3}$. [1]

Zvýšená koncentrácia vo vnútorných priestoroch budov zvyšuje množstvo nádorových ochorení obyvateľstva. Zvýšenie koncentrácie radónu o $100\ Bq\ m^{-3}$ môže zvýšiť počet prípadov rakoviny pľúc až o 16 %. [11]

1.2 Radónová problematika v pracovných priestoroch

Riešeniu radónovej problematiky v pracovnom prostredí a ochrane zdravia pracovníkov sa začína venovať zvýšená pozornosť začiatkom päťdesiatych rokov, keď nastal rozvoj uránového priemyslu. Došlo k nasadeniu veľkého počtu baníkov do uránových baní a k obavám zodpovedných pracovníkov zo zvýšeného počtu profesionálnych rakovín pľúc. Začalo sa s intenzívnejším vetraním, prebiehali merania výskytu radónu a zavádzali sa metódy osobnej dozimetrie. Neskôr sa pozornosť preniesla aj na neuránové bane a jaskyne.

V krasových jaskyniach je situácia s ochranou pracovníkov iná, kedže v dôsledku zavedenia zvýšeného ventilačného systému môže dochádzať k narúšaniu krasových útvarov. V tomto prípade sa odporúča pre pracovníkov regulovanie počtu vstupov a tým

znižovanie počtu hodín strávených v jaskynných priestoroch. V súčasnosti mnohé krajiny majú prijaté legislatívne opatrenia na ochranu pracovníkov, ktoré nadväzujú na medzinárodné odporúčania.

Na oddelení radiačnej hygieny SZU v Bratislave [10] sa začali venovať výskytu radónu v jaskynných priestoroch koncom r.1992, kedy Ministerstvo kultúry SR, pod ktoré vtedy Správa slovenských jaskýň spadala, požiadalo o spoluprácu. Začalo sa so sledovaním sezónnych variácií výskytu radónu pomocou DSPF (Detektor stôp v pevnej fázi) v desiatich turistických krasových jaskyniach.

Absorbované dávky žiarenia pracovníkov v jaskyniach môžu dosahovať ozaj vysoké hodnoty. Napríklad v období v roku 1995 až 1996 bola zaznamenaná v Ochtínskej aragonitovej jaskyni celoročná dávka jedného pracovníka 37,5 mSv. [12]

Na základe výsledkov meraní a odporúčnia pracovníkov Štátneho zdravotného ústavu v Banskej Bystrici, od roku 1998 sa prešlo k sledovaniu radiačnej záťaže vo všetkých krasových jaskyniach spadajúcich pod Správu slovenských jaskýň v Liptovskom Mikuláši. Osobnému monitorovaniu podliehali všetci stáli pracovníci (turistickí sprievodcovia) a osobný dozimeter bol pridelený aj každému brigádnikovi. Následne, v roku. 2000 vstúpil do platnosti zákon č.470/2000 Z.z., ktorý je novelou zákona č.272/1994 Z.z. o ochrane zdravia ľudí, a v januári r.2001 vyhláška MZ SR č.12/2001 na základe ktorých sú priestory jaskýň klasifikované ako pracoviská so zdrojmi ionizujúceho žiarenia so zvýšeným výskytom prírodných zdrojov žiarenia a podliehajú osobnému monitorovaniu pracovníkov. V roku 2001 bol zriadený Centrálny register dávok pracovníkov so zdrojmi ionizujúceho žiarenia na Úrade verejného zdravotníctva Slovenskej republiky a začalo sa so systematickým sledovaním veľkosti ožiarenia všetkých pracovníkov so zdrojmi ionizujúceho žiarenia. SR bola zapojená do projektu Európskej komisie ESOREX [33].

V súčasnosti, v súlade s platnou legislatívou (Zákon č. 355/2007, Nariadenie vlády Slovenskej republiky č. 345/2006, Nariadenie vlády SR č. 346/2006 Z.z., Vyhláška MZ SR č. 545/2007) je osobná dozimetria pracovníkov zabezpečovaná na Oddelení radiačnej hygieny SZU v Bratislave v 11-tich jaskyniach SR a pravidelne je monitorovaných cca 120 pracovníkov ročne. Monitorovanie prebieha v štvrťročných intervaloch a pre stanovenie OAR na pracovisku sa využívajú DSPF typu CR-39, ktoré sú umiestnené v difúznej komôrke (Obrázok 1.1). Pracovníci ich nosia do podzemných priestorov.

Výsledky stanovenia osobných dávok pracovníkov v podzemí sú zasielané do Centrálneho registra dávok. Kontroluje sa či zamestnanec neprekročil zákonom stanovené limity ožiarenia na pracovníka z rôznych oblastí povolania.

Obrázok 1.1 Difúzna komôrka s osobným dozimetrom.

1.3 Meranie ožiarenia pracovníka radónom pomocou detektoru stôp

V tejto časti popíšem aktuálny stav vyvíjaného meracieho systému s chemicky leptaným stopovým dozimetrom v laboratóriu na Oddelení radiačnej hygieny SZU v Bratislave.

1.3.1 Detekčný materiál

Ožiarenie pracovníka radónom sa meria pomocou pasívneho detektoru stôp v pevnej fázi [3], ktorý je počas merania uložený v špeciálnom držiaku nosenom na oblečení. Tento držiak na detekčný materiál nosí pracovník v priestoroch, kde je predpokladaná zvýšená radiačná záťaž. Metóda je používaná, pretože je nenákladná. Laboratórium, pre ktoré je riešené zadanie, používa materiál CR-39.

Materiál pozostáva z polymérových reťazcov alyl diglykol karbonátu, ktoré sa pri zásahu alfa časticou poškodzujú a teda vznikajú nanoskopické poškodenia materiálu. Po expozičnej dobe sa dozimeter leptá chemicky v leptacom roztoku s lúhom sodným, čím sa malé defekty molekúl zväčšia na rozmery pozorovateľné mikroskopom. [8] Poškodené časti polymérovej dosky sa totiž odleptávajú podstatne rýchlejšie, ako nepoškodený materiál. [4] [5] [6] [13]

Materiál nieje citlivý na beta, röntgenové žiarenie ani na gamma žiarenie [7], čím sa vylučuje možnosť sledovať niektoré zdroje žiarenia v medicínskom prostredí, avšak je stále použiteľný napríklad v jaskynných priestoroch.

Stopa, ktorá vznikla preletom častice približne kolmo na dosku je, po procese leptanania, vidieť pod mikroskopom ako bublinka, ktorá na povrchu zodpovedá

polguľovému tvaru odleptaného materiálu. Stopa, ktorá vletela skoro vodorovne do dosky sa prejaví ako súvislá čiastočne zakrivená rada bubliniek. Medzi dávkou žiarenia, ktorú pracovník absorboval a počtom stôp viditeľných na detektore existuje úmera, ktorá sa zistí kalibráciou, ktorej výsledkom sú koeficienty do polynomiálnej rovnice. Najsilnejšou zložkou rovnice je lineárny koeficient.

Metóda merania detektorom stôp v pevnej fázi je integrálna. Pomocou nej sa dá určiť priemerná koncentrácia radónu v priestore, kde sa pohyboval pracovník a jeho absorbovaná dávka žiarenia. Meranie musí prebiehať dlhšie.

Na začiatku si firma objedná na jeden kvartál istý počet dozimetrov podľa toho koľko pracovníkov sa bude pohybovať v podzemných priestoroch. Dozimetre sa v laboratóriu narežú, označia sériovými číslami a zabalia sa do alobalu aby sa chránili pred expozíciou počas transportu. Jeden dozimeter je pozaďový a je ponechaný na stálom mieste, spravidla v miestnosti správcu jaskyne, ktorá nieje v podzemných priestoroch. Doba expozície je potom doba, za ktorú bol exponovaný pozaďový detektor (od rozbalenia, po zabalenie). Slúži korekcia na pozadie pri výpočte dávky. Ak je osobný dozimeter pridelený pracovníkovi hneď na začiatku, (teda je rozbalený v rovnakom čase ako pozaďový), doba expozície oboch dozimetrov je rovnaká. Ak mu je pridelený neskôr, doba expozície predstavuje čas, kedy bol odbalený a zabalený a počet stôp z pozadia pre korekciu sa určí pomerne k počtu stôp pozaďového detektoru a dobe expozície osobného detektora. Predpokladom je, aby detektor, ak ho pracovník mená u seba pri vstupe do jaskyne, bol odložený na tom istom mieste ako pozaďový.

Pred odoslaním dozimetrov do laboratória sa znova zabalia do alobalu. V laboratóriu sa dozimetre leptajú ponorené v 30% roztoku NaOH po dobu 18 hodín pri teplote 70 °C. Táto teplota je presne udržiavaná, pretože zmena iba o 1°C spôsobí zmenu veľkosti leptaných bublín až o 15%, čo by zhoršilo opakovateľnosť merania. Vyššia teplota by spôsobila rozleptanie dozimetrov. Ihneď po uplynutí doby leptania sa detektory opláchnu destilovanou vodou a následne 50% roztokom kyseliny citrónovej.

Vyhodnocovanie integrálneho osobného dozimetra spočíva v stanovení expozície, t.j. časového integrálu objemovej aktivity radónu (vzťah 1)

$$I_{OAR} = \int_0^T c_A(t).dt \ \left[\frac{Bq\ h}{m^{-3}}\right] \qquad (1)$$

Pri stanovení časového integrálu pasívneho osobného dozimetra v pracovnom priestore I_{OS} je potrebné poznať jeho expozíciu v priestore uloženia mimo pracovnej doby I_{KON} a preto sa s osobnými dozimetrami zasiela aj pozaďový dozimeter, ktorý je umiestnený spravidla v kancelárii správcu jaskyne, teda nie v podzemí a výsledná expozícia sa vypočíta podľa vzťahu 2.

$$I_{OS} = I_{DOZ} - I_{KON}.\left(1 - \frac{T_{OS}}{T}\right)\left[\frac{Bq\ h}{m^{-3}}\right] \qquad (2)$$

T_{OS} je čas expozície osobného dozimetra a Tje čas expozície pozaďového dozimetra. I_{OAR} je integrálna objemová aktivita radónu v miestnosti. I_{DOZ} je integrálna objemová aktivita radónu pre dozimeter a I_{KON} je je integrálna objemová aktivita radónu pre pozaďový dozimeter.

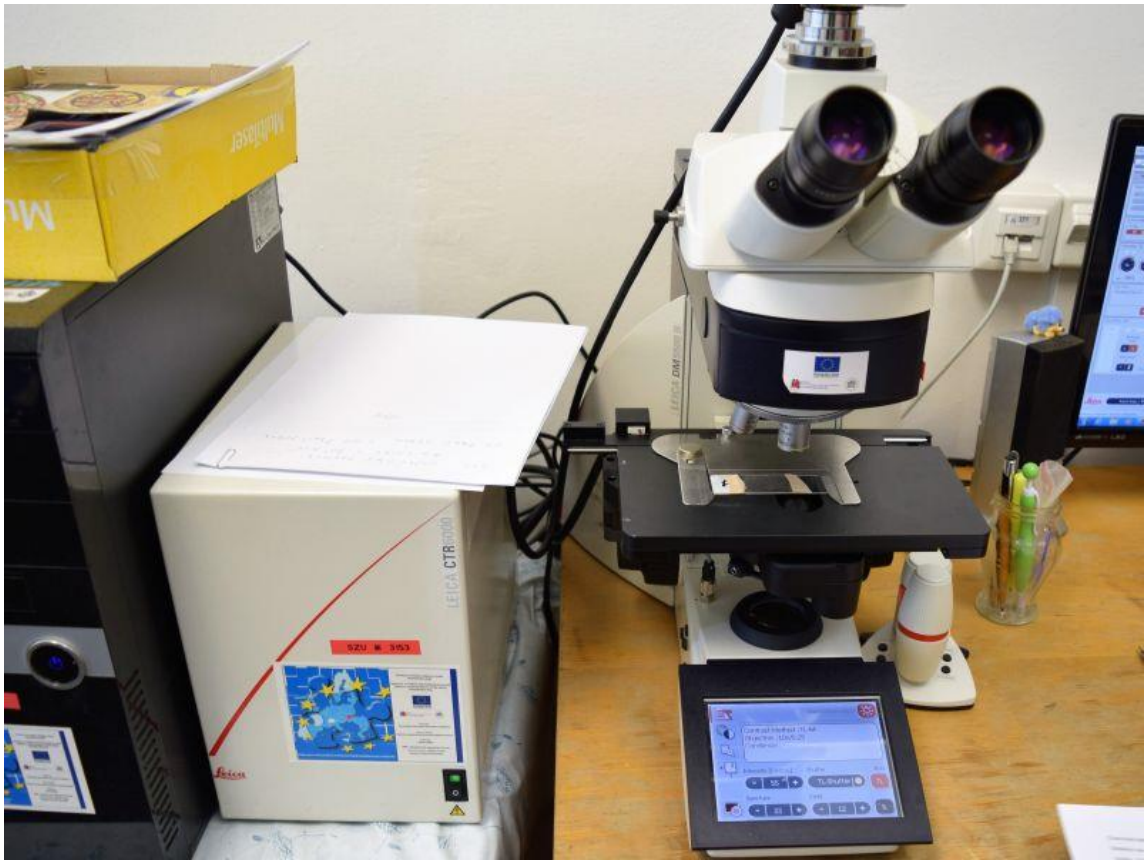
Integrál OAR sa vypočíta na základe počtu stôp stanovených pomocou obrazového analyzátora stôp Quantimet 520 a účinnosti. Pre zviditeľnenie stôp na exponovanom dozimetri je použitá už popísaná metóda chemického leptania a účinnosť je stanovená vždy pre sadu, z ktorej bol detektor vyrezaný. Účinnosť sady je stanovená na základe kalibrácie náhodne vybraných detektorov z jednotlivých platní sady a uskutočňuje sa v radónovej komore EISPP v Štátnom metrologickom stredisku pre radónové veličiny, ktoré je autorizované Úradom pre normalizáciu, metrológiu a skúšobníctvo SR v Bratislave na overovanie radónových veličín.

V laboratóriu sa na snímanie dozimetrického skla používa mikroskop DM5500 B od firmy Leica. [19] Tento model je dobre vybavený pre snímanie stôp na dozimetri, pretože má motorizovaný posun meraného sklíčka, ktorý sa dá ovládať počítačom pomocou makro programu, a takisto má dokúpený modul kamery Leica DFC 295, ktorá prenáša obraz do PC. Pre zaostrovanie obrazu a posun stolčeka mikroskopu sa používa špecializovaný ovládač s troma veľkými otočnými gombíkami. Pomocou dvoch gombíkov sa posúva stolček v osách X a Y a tretím gombíkom sa zaostruje snímka. Model ovládača je Leica Smart Move Controller. [20]

Makrá sa píšu v prostredí Leica Application Suite Macro Editor, ktoré ponúka základné funkcie pre automatické ovládanie mikroskopu ako sú grafické prvky rozhrania, ktoré vedú užívateľa pri rutinnej práci merania, zmienený posun snímkovaným sklíčkom, vytváranie snímok a ich ukladanie do medzipamäti alebo na disk, základné obrazové operácie na snímkach v medzipamäti ako je úprava jasu, kontrastu, rôzne operácie sčítania a odčítania obrazu, a nakoniec obsahuje podporu pre rozpoznávanie objektov v obraze a to pomocou morfologických operácií, akými sú napríklad erózia, dilatácia, open, close.

Pre meranie je dôležitá funkcia rozčlenenie meranej plochy na mriežku 8x8, z toho dôvodu, že sa pri snímkovaní objektívom mikroskopu nezaberie celá plocha dozimetra. Takto sa zvolená meracia plocha rozčlení na 64 segmetov, v ktorých softvér LAS umožňuje nastaviť zaostrenie obrazu.

Dôvod, prečo sa vytvára nový softvér a hľadá sa alternatívna detekcia stôp v obraze je ten, že pôvodná detekcia tohto novo nasadzovaného mikroskopu sníma príliš malé množstvo skutočných stôp v obraze a taktiež napočíta veľké množstvo rôznych defektov na povrchu dozimetrického sklíčka ako sú škrabance spôsobené čistením dozimetra od diamantového prachu. Taktiež sú napočítavané aj rozmazané stopy na pozadí obrázku, ktoré neboli zaostrené mikroskopom, a preto sa nedajú spoľahlivo počítať. Tým pádom má pôvodná metóda nízku citlivosť a vysokú chybu merania.



Obrázok 1.2 Zariadenie na vyhodnocovanie detektorov stôp v pevnej fáze, Quantimet. Mikroskop Leica DM5500 B je nastavený na zväčšenie 10x a je v ňom vložené dozimetrické sklíčko. Vpravo sa nachádza ovládač, ktorým sa zaostrujú snímky a manuálne sa nastavuje motorizovaný stolček.

1.3.2 Postup snímkovania mikroskopom

Snímkovanie v laboratóriu pomocou vyvíjaného systému prebieha vložením vyleptaného sklíčka do motorizovaného držiaku a spustením makra v programe LAS.

Skript najprv umožní laborantovi ručne nastaviť zaostrenie všetkých 64 snímok, pričom automaticky posúva vzorku pod mikroskopom. Interakcia užívateľa je riešená cez panel zabudovaný priamo do aplikácie LAS, takže sa využíva pracovný proces zamýšľaný dodávateľom mikroskopu. Zaostrovanie predstavuje časovo náročnú operáciu. Po zistení nastavenia zaostrenia skript urobí 64 snímok v mriežke 8x8. Každá snímka je veľkosti 2048 x 1536 pixelov (šírka x výška), má 24 bitovú farebnú hĺbku a má formát jpeg. Následne sa spočíta počet stôp vo všetkých snímkach pomocou algoritmu Watershed, kde sa používajú vnútorné funkcie prostredia LAS Macro.

Pracovný postup pri snímkovaní mikroskopom:

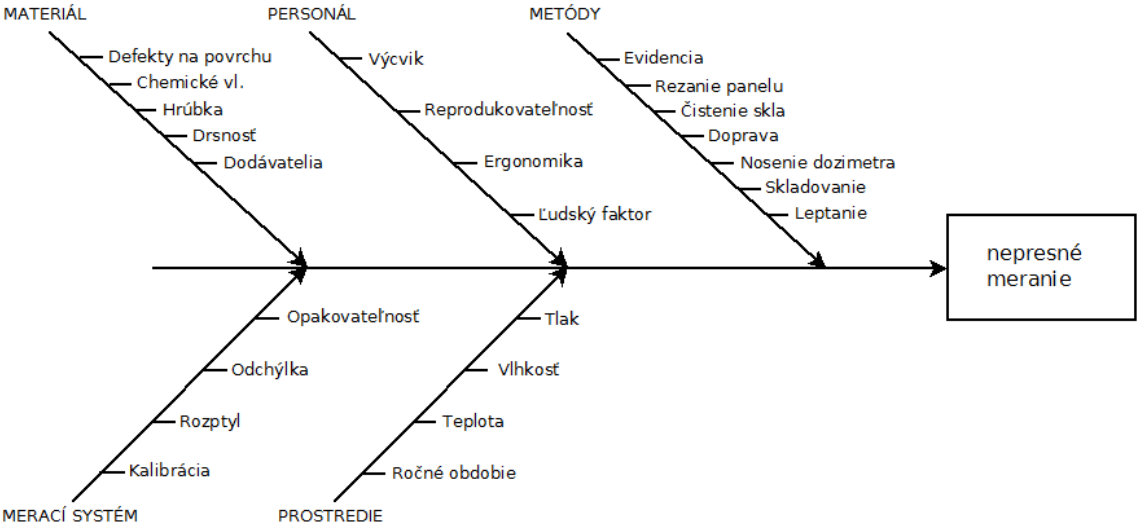
- Užívateľ vyberie druh merania.
- Užívateľ zadá názov merania.
- Vytvorí sa adresár s názvom merania, ktorý sa použije na ukladanie súborov merania.
- Užívateľ najde ľavý horný okraj na sklíčku, od ktorého bude snímkovaná meraná plocha.
- Následne zabudovaná podpora v programe LAS posúva sklíčko po jednotlivých segmetoch a užívateĺ podľa empirickej skúsenosti zaostruje jednotlivé snímky približne na povrch sklíčka, nastavenie zaostrenia pre snímky sa zapamätá pre ďalší krok.
- Skript automaticky prejde všetkých 64 segmentov na sklíčku a z každého segmentu vytvorí snímku. Tieto snímky sa ukladajú na HDD vo formáte jpg. Toto je ďalší časovo náročný krok merania.
- Následne skript načítava snímky z HDD a vykonáva počítačové spracovanie obrazu, kde napočítava počet detegovaných stôp na snímke. Tento krok je tiež časovo náročný z dôvodu výpočtovej náročnosti rozpoznávania obrazu.
- Uloženie protokolu s výsledkami merania vo formáte xls kompatibilným s verziou Excel 97, ukladá sa celkový počet detegovaných stôp a počet stôp detegovaných v jednotlivých segmentoch.
- Užívateľ prepočíta pomocou výpočtu a koeficientov, ktoré boli získané kalibráciou meracieho systému počet detegovaných stôp na dávku žiarenia, ktorú absorboval pracovník, ktorý nosil dozimeter.

2 ANALÝZA SYSTÉMU MERANIA

V tejto kapitole je detailnejšie popísaný merací systém, a rôzne vplyvy metód spracovania materiálu a vplyvy prostredia, ktoré pôsobia na mieru nepresnosti merania.

2.1 Fishbone diagram

Vplyvy na presnosť merania budú posúdené Išikawovym diagramom príčin a následkov. [21]



Obrázok 2.1 Ishikawov diagram popisujúci vplyvy na presnosť merania

2.2 Materiál

Materiál v podobe panelov z polyméru CR-39 je do laboratória dodávaný rôznymi výrobcami a preto panely prešli rôznymi výrobnými procesmi a môžu sa líšiť rôznymi vlastnosťami. Konkrétne už od výroby majú panely istú nerovnomernosť tvaru povrchu, a z tohto dôvodu je obsluha mikroskopu nútená preostrovať jednotlivých 64 segmentov pri snímkovaní dozimetra do správnej hĺbky pri povrchu materiálu. Ďalej je dôležité aby povrch nebol hrbolatý, pretože sa to prejaví ako viditeľné tmavé objekty na snímke, ktoré by rušili detekciu obrazu. Tým pádom povrch panelu nesmie prekročiť istú hodnotu drsnosti. Materiál by mal mať minimálne mechanické poškodenie z toho istého dôvodu, veľké škrabance vrhajú na mikroskope tiene, ktoré sa prejavujú ako viditeľné tmavé objekty.

2.3 Personál

Výcvik laboranta pri meraní je dôležitý pre dosiahnutie presnosti, reprodukovateľnosti a opakovateľnosti merania. Laborant do merania zasahuje pri rezaní panelov na jednotlivé sklíčka dozimetrov, pri ich vyleptávaní, pri ich snímaní na mikroskope, pri manuálnej korekcii vyhodnocovania obrazu algoritmom a nakoniec matematickým výpočtom pri konečnom stanovení dávky osoby, ktorá dozimeter nosila.

Metóda je dobre reprodukovateľná ak viacero rôznych laborantov zmeria veľmi podobné hodnoty počtu detegovaných stôp pre rovnaký pôvodný dozimeter. To by sa dalo pri otestovaní postupu od kroku s mikroskopom a ďalej overiť experimentom s viacerými prizvanými dobrovoľníkmi, čo si meranie vyskúšajú po zaškolení. Rozdiely v hodnotách budú spôsobené rôznym nastavením hĺbky zaostrenia mikroskopu pri 64 rôznych snímkach a pri ručnom dolaďovaní výsledku rozpoznania obrazu. Počet nameraných stôp má veľkú variabilitu z dôvodu stochastickosti procesu rádioaktívneho rozpadu, a preto by bolo nepraktické robiť pokus reprodukovateľnosti napr. meraním koncentrácie v jednej miestnosti, pretože by musela obsahovať veľké množstvo dozimetrov, pre každého laboranta jeden a dozimetre by nemohli byť všetky umiestnené úplne na rovnakom mieste, takže by mali rozdielne podmienky prúdenia vzuchu, v ktorom sa nachádza radón. Taktiež by museli byť všetky merané v rovnaký čas z dôvodu sezónnosti aktivity radónu v bytových priestoroch. [22]

Ergonomickosť softvéru má spočívať hlavne v rýchlosti spracovania jedného dozimetra. Celé meranie na mikroskope vrátane rozpoznávania obrazu spolu s manuálnou korekciou by nemalo zabrať viac ako 15 minút z dôvodu, že do laboratória nárazovo prichádza veľké množstvo vzoriek. Každá ušetrená minúta behu softvéru výrazne zlepší súčasné podmienky na prácu. Ak bude dávať softvér dostatočne presnú detekciu, potom by nemuseli byť manuálne korekcie vôbec nutné. Okrem rýchlosti je dôležitá jednoduchosť ovládania. Preto softvér musí zapadať do súčasného pracovného postupu a mal by automatizovať čo najviac krokov. Napríklad makro ktoré snímkuje pomocou mikroskopu vytvára adresár s meraním a do neho postupne uloží obrázky a posledný obrázok, ktorý vytvorí má názov IMG64.jpg. Podľa prítomnosti nového adresára s novým dátumom by sa dalo užívateľovi automaticky ponúknuť otvoriť súbory na spracovanie bez toho, aby musel zbytočne hľadať adresáre na disku PC. Takisto by do softvéru bol ideálne zahrnutý aj samotný prepočet stôp na dávku pracovníka, ktorému bol dozimeter pridelený, ktorého výsledok by sa uložil do súboru s reportom merania.

2.4 Metódy

Pri rezaní panelu sa panel znečistí drobnými úlomkami materiálu a pri následnej snahe tieto úlomky odstrániť sa povrch odrezaného sklíčka poškriabe. Tieto škrabance majú pod mikroskopom podobu tmavých škvŕn a spôsobujú problémy pri detekcii pomocou pôvodného algoritmu.

Sklo sa čistí od úlomkov pomocou špeciálnej jemnej handričky, ktorá zanecháva menej škrabancov. Bolo skúšané aj použitie stlačeného vzuchu, avšak materiál praskal po rýchlom schladení expandujúcim stlačeným vzduchom.

Leptá sa presne definovaný čas pomocou presne určenej koncentrácie NaOH vo vode. Leptanie môže zhoršovať rovnomernosť povrchu a vplývať na ostrenie na povrch sklíčka. Ďalej sa leptaním odleptá najvrchnejšia vrstva materiálu a tým sa odleptávajú stopy, ktoré letia skoro rovnobežne s povrchom materiálu. [13] Tieto stopy sa teda nedajú detegovať. Z praktických dôvodov sa nesnímkujú stopy, ktoré sú hlbšie v materiáli, pretože každý segment dozimetra sa fotografuje iba raz a pri jednom nastavení zaostrenia.

Pri skladovaní panelov sa môžu ožiariť alfa žiarením a tým sa exponujú. Preto ich treba tieniť.

Pre meranie dozimetrami ich treba dopraviť na dané pracovisko, pri čom sa po ceste čiastočne ožiaria. Aby bola táto chyba oproti zmeraným hodnotám zanedbateľná, musí byť doba dopravy výrazne kratšia ako doba expozície pracovníkom.

2.5 Merací systém

Opakovateľnosť merania sa vyšetruje zložitejšie z dôvodu stochastickej povahy rádioaktívnych rozpadov. 10 rôznych dozimetrov, umiestnených v rovnakom priestore, bude zasiahnutých iným počtom alfa častíc a bude vyleptaný iný počet stôp. Navyše v priebehu celého roka sa dá vysledovať meniaci sa pravidelný trend aktivity rádioaktivity v uzavretých priestoroch. [22] Z toho dôvodu, ak sa pracovník vyskytuje v rovnakom priestore, nameria iné hodnoty aktivity v iný mesiac v roku. Ďalej pracovník sa môže pohybovať v rôznych priestoroch v rôzne pravidelných harmonogramoch alebo dochádzať do práce iný počet hodín a tak môžu v rôznych meracích obdobiach vychádzať rozdielne hodnoty pre toho istého zamestnanca.

Odchýlka pri meraní môže byť spôsobená nepresnou kalibráciou. Taktiež rozptyl hodnôt je hlavne spôsobený stochastickou povahou rádioaktivity. Ďalej môže byť hodnota ovplyvnená metódou leptania, čistenia sklíčka. Najväčší vplyv meracieho systému budú mať manuálne nastavenie polohy na sklíčku, ktorá bude snímkovaná, manuálne zaostrovanie do určitej hĺbky v materiáli a tiež manuálna korekcia výsledkov rozpoznania obrazu. Naopak použitý algoritmus na rozpoznanie obrazu by nemal mať veĺký vplyv na rozptyl z dôvodu deterministickej povahy algoritmov rozpoznávania

obrazu vrátane algoritmu Watershed aj Houghovej transformácie. Predspracovanie obrazu ako je rozmazanie má takisto deterministickú povahu.

Kalibrácia dozimetrov sa vykonáva v kalibračnej komore, ktorá má pevné rozmery a stanovený objem vzduchu vnútri komory. Do komory sa vpúšťa radón produkovaný roztokom s rádiom, ktorého hmotnosť bola presne určená. V komore sa tak nachádza presná koncentrácia radónu na meter kubický a dozimeter je ožarovaný stanovenú dobu kalibrácie. Po vyleptaní sa na dozimetroch objavia stopy spôsobené bombardovaním alfa časticami. Objemová aktivita radónu v komore je spočítaná a teda známa, vďaka čomu sa dá určiť vzťah medzi počtom vyleptaných častíc na kalibrovaných dozimetroch a objemovou aktivitou radónu v komore.

2.6 Prostredie

Na meranie môžu vplývať fyzikálne vlastnosti ako je teplota, vlhkosť tlak, a veľký vplyv má aj čas merania, tj. mesiac v roku. Fyzikálne vlastnosti vzduchu môžu vplývať na aktuálnu aktivitu radónu v miestnosti. Obdobie roku silno vplýva na to v akých množstvách vyviera radón z geologického podložia. [22] Je možné, že v zimnom období je nameraná dávka v miestnosti vyššia, pretože sa menej často vetrajú vnútorné priestory budov.

3 ALGORITMY NA POČÍTANIE OBJEKTOV V OBRAZE

V tejto kapitole bude popísaný algoritmus watershed, ktorý bol použitý v pôvodnom makre LAS pre nový mikroskop v laboratóriu, ktoré malo byť použíté na počítanie stôp. Ďalej bude popísaná Houghova transformácia, ktorá je v tejto práci testovaná pre použitie v novom softvéri.

3.1 Algoritmus watershed

Tento algoritmus segmentuje obraz, ktorý bol vopred prevedený na binárnu bitovú mapu, na jednotlivé oddelené objekty. Výsledné masky prekrývajúce nájdené objekty majú tvar machule. Následne sa dajú pri jednotlivých segmentoch zmerať parametre, ako je stred objektu, veľkosť vzniknutej plochy, kruhovitosť, pomer strán a podobne. [14]

Metóda pri segmentácii nerozlišuje medzi tvarmi objektov ale pomáha jej ak sú objekty kruhovité a ak niesú príliš prekryté, alebo aby objekty neboli pospájané hrubými prvkami v obraze. Hľadané objekty sa rozlíšia kontrolou, či sú zmerané parametre ako je kruhovitosť alebo plocha v určenom rozmedzí hodnôt. Obrázok 6.4 ilustruje výstup z tejto metódy.

3.2 Pojmy operácií pre segmentáciu obrazu

Prostredie LAS Macro Editor obsahuje operácie na úpravu obrazu, ktoré budú vysvetlené v tejto podkapitole. Popis príkazov pochádza z nápovedy LAS Macro Editora. Príkazy sa používajú v pôvodnom algoritme.

3.2.1 Detect

Príkaz umožňuje konvertovať pôvodný farebný obrázok na čiernobiely s jednobitovou hĺbkou. Výsledný pixel na výstupnom binárnom obrázku je buď čierny alebo biely.

Syntax varianty príkazu použitého v pôvodnom skripte:

Detect (blacker than LowThresh, from InputImage into OutputBinary)

Príkaz nastaví na výstupe pixely na bielu farbu, ak pôvodná hodnota pixelu bola tmavšia ako hodnota spodného limitu LowThresh. V podstate dochádza aj ku inverzii farieb. Ostatné pixely sú čierne. InputImage značí ktorý vstavaný obrázkový buffer sa použije ako vstupný obrázok a OutputBinary značí, ktorý vstavaný binárny obrázkový buffer sa použije ako výstup operácie.

3.2.2 Amend

Príkaz *Amend* umožňuje použit na obraz niekoľko morfologických operácii, ktoré slúžia na oddeľovanie alebo kombinovanie súčastí nachádzajúcich sa v obraze. Operácie pracujú postupne s malým jadrom istého tvaru, ktoré určuje, s ktorými pixelami sa bude pracovať, toto jadro určuje susedstvo niekoľkých pixelov. V pôvodnom skripte sa používalo nastavenie *Disc*, ktoré aproximuje tvar kruhu.

Použité operácie s príkazom *Amend* v pôvodnom algoritme sú:

- *Erode* (erodovať) – vymieňa v rámci susedstva hodnoty sivých pixelov minimálnou hodnotou v susedstve. Tmavé objekty v obraze sa zmenšujú. [16]
- *Dilate* (dilatovať) – vymieňa v rámci susedstva hodnoty sivých pixelov maximálnou hodnotou v susedstve. Tmavé objekty v obraze sa zväčšujú. [15]
- *Open* – opakovane vykonáva akciu *Erode*, za čím nasleduje opakovane akcia Dilate. Následkom je vyčistenie obrazu od malých tmavých objektov. [17]
- *Close* – opakovane vykonáva akciu *Dilate*, za čím nasledujee opakovane akcia Erode. Obraz sa vyčistí od malých svetlých objektov. [18]

3.2.3 Identify - FillHoles

Ďalší príkaz, ktorý je použitý v makre je príkaz Identify, ktorý obsahuje užitočnú operáciu FillHoles, ktorá vypĺňa diery v obraze.

3.2.4 Segment

Príkaz segment je používaný na rekonštruovanie hraníc rôznych objektov v obraze. Oddeľuje napríklad od seba čiastočne prekrývajúce sa objekty, tak aby sa dali spočítať a merať ich plocha, pomery strán a podobne.

Ako je popisované v manuáli od LAS Macro Editor-u, jeho použitie má svoje limitácie. Operácia segmentácie je binárna operácia a k svojej práci používa iba binárny obrys pôvodného objektu. Z toho dôvodu samotná operácia threshold musí viesť ku kvalitnému kruhovému obrysu detegovaných stôp. Ďalej je výsledný binárny obrys objektu ovplyvnený tým, ako sa objekt rozpadne opakovanými operáciami erózie. (Open)

Ďalej má na výsledok vplyv, aký tvar jadra sa použije pri detekcii, na detegovanie kruhových tvarov sú najlepšie jadrá *disc* (podobný kruhu) a *octagon.* (osemuholník) Tiež čím je objekt zložitejšieho tvaru, tým bude segmentácia menej účinná. Problém prestavujú pozdĺžne a tenké objekty.

Parameter *Filter* nastavuje ako hrubo sa obraz segmentuje, respektívne veľkosť okna, ktoré sa použije v jednotlivých krokoch segmentácie. Čím je hodnota väčšia, tým sa hľadajú väčšie súčasti obrazu. Zvyšovaním tejto hodnoty sa zmenšuje nadsegmentácia,

ale môže ostať spojených viacej objektov dokopy. Nastavenie tohoto filtra je pravdepodobne závislé na rozmere objektu a preto je ťažké ho správne nastaviť pre detegovanie veľkých a zároveň malých stôp na detektore.

Ďalším parametrom segmentácie je *MaxCycles*, ktorý ovplyvňuje počet nádjených segmentov v obraze a nastavuje sa empiricky. Príliš nízka hodnota spôsobuje že objekty ostanú spojené a bude ich nájdený menší počet. Príliš vysoký hodnota spôsobí nadsegmentáciu, kde budú jednotlivé objekty rozkúskované na viac častí a napočítané zvlášť. Optimálna hodnota parametru má byť o trochu vyššia ako dostačujúca hodnota, ktorá však nespôsobí nadsegmentáciu.

3.3 Popis pôvodného algoritmu

Pôvodný algoritmus napísaný v prostredí LAS Macro Editor a využíva operácie na segmentáciu obrázku a následne operácie na detegovanie jednotlivých objektov a meranie ich vlastností. Počet objektov, ktoré boli akceptované s vyhovujúcim tvarom a plochou je výsledný počet nájdených stôp. Používa sa nasledujúca postupnosť operácií a podobá sa tým algoritmu známemu pod názvom watershed:

- Prečítanie obrazu zo súboru.
- Detekcia tmavších pixelov ako hodnota 115, budú nastavené na bielu farbu, ostatné budú čierne. Vzniká binárny obrázok.
- Amend Close, počet cyklov 15, operátor disc, vypnutá erózia okrajov obrazu
- Identify FillHoles
- Amend Open, počet cyklov 5, operátor disc, vypnutá erózia okrajov obrazu
- Segment, parameter *filter* nastavený na 10, max cycles nastavené na 25, operátor disc.
- Measure Frame súradnice x,y sú 50, 100, šírka a výška obdĺžnika s meranými objektami je 1948 x 1393.
- Measure Feature, ferets je nastavene na 64, minimálna plocha detegovaného objektu je nastavená na 10.
- Feature Accept – akceptujú sa súčasti, ktoré majú plochu od 150 do 8000 a aspect ratio (pomer strán objektu) je od 1 po 3 (umožňuje detegovať okrem kruhov aj elipsy)
- Detegovaný počet stôp na snímke je rovný počtu detegovaných a akceptovaných súčastí rozsegmentovaného obrazu.

3.4 Houghova transformácia

Tento algoritmus miesto všeobecných objektov, ako pri algoritme watershed, deteguje objekty, ktoré majú geometrické tvary, existuje napríklad modifikácia pre detegovanie čiar, a tiež modifikácia na detekciu kruhových tvarov. Výstupom algoritmu je pozícia x,y detegovaného kruhu v pixeloch a veľkosť polomeru kruhu v pixeloch. [24]

3.4.1 Algoritmus detekcie stôp

Uvedený je typický postup pri použití houhovej transformácie z knižnice OpenCV pri detekcii kruhov pre použitý programovací jazyk Python:

- Prečítanie obrazu zo súboru
- Rozmazanie obrazu (funkcia *cv2.medianBlur*)
- Konverzia snímky na odtiene sivej (funkcia *cv2.cvtColor*)
- Houghova transformácia pomocou funkcie cv2.HoughCircles
- Výstupom je zoznam nájdených kruhov obsahujúci stred kruhu a polomer.

Ako by sa ďalej postupovalo pri detekcii stôp bude popísané v kap. Navrhované riešenie problému.

3.4.2 Knižnica OpenCV

Knižnica OpenCV (Open source Computer Vision) [28] obsahuje množstvo funkcií, ktoré sú zamerané na počítačové videnie a spracovanie obrazu. Je napísaná v jazykoch C a C++ a je optimalizovaná na rýchlosť pre aplikácie v reálnom čase. Knižnica obsahuje funkcie na filtráciu obrazu ako je zaostrenie, rozmazanie obrazu, erózia alebo dilatácia. Ďalej obsahuje funkcie na konverziu sivého obrázku na čiernobiely (thresholding), štruktúrnu analýzu obrazu ako je vyhľadávanie kontúr v obraze, analýzu pohybu a sledovanie pohybujúcich sa objektov a detegovanie súčastí alebo objektov v obraze.

Pre túto prácu je zaujímavá podpora detekcie súčastí v obraze. Knižnica vie detegovať čiary a kruhy. V tejto práci používam druhú verziu knižnice. Pre účely skúšania algoritmu Houghovej transformácie sú vytvorené skripty v jazyku Python, a tak budú popísané volania tejto knižnice v Pythone.

3.4.3 Funkcia HoughCircles

Táto funkcia implementuje Houghovu transformáciu a volá ďalšie funkcie pred samotnou transformáciou na zlepšenie detekcie kruhov. [25] Implementuje algoritmus označený ako 21HT v publikácii [23]. Vnútorne sa v tejto funkcii tiež používa funkcia *Canny* z knižnice OpenCV, ktorá hľadá hrany objektov v obraze pomocou algoritmu popísaného v publikácii [26].

Takto vyzerá volanie transformácie v jazyku python pri použití všetkých parametrov [27]:

cv2.HoughCircles(image, method, dp, minDist, circles, param1, param2, minRadius, maxRadius)

Parameter image je 8 bitový obrázok s jedným kanálom s farbami v odtieňoch sivej farby.

Ako metóda je nastavená vždy možnosť cv2.HOUGH_GRADIENT, pretože knižnica zatiaľ implementuje iba typ algoritmu 21HT. [23]

Parameter dp určuje pomer veľkosti akumulátorovej matice, ktorá sa používa pri hľadaní kruhov pomocou Houghovej transformácie. Pri hodnote 1 je šírka a výška akumulátora zhodná s rozmermi spracovaného obrázku. Pri hodnote 2 má akumulátor polovičnú šírku a výšku oproti obrázku.

Parameter minDist určuje minimálnu vzdialenosť stredov dvoch kruhov v pixeloch, ktoré budú detegované oddelene od seba. Ak je parameter príliš nízky, môže sa metódou detegovať jeden kruhový objekt viacnásobne ako viacero kruhov blízko seba. Naopak ak je parameter príliš vysoký, niektoré kruhové objekty, ktoré sú príliš blízko seba nebudú detegované.

Parametre param1 a param2 nastavujú detekciu hrán v obraze, ktorá je robená pomocou volania funkcie *Canny*. Param1 je hraničná hodnota pre nastavenie vnútornej hysterézie funkcie Canny. Param2 je hraničná hodnota pre stredy kruhov v detekčnom kroku metódy. Zmenšovanie tohto parametru môže vyvolať zvýšenie počtu falošných detekcií.

Parametre minRadius a maxRadius určujú min. polomer a max. polomer v pixeloch kruhu, ktorý bude detegovaný. Príliš nízka hodnota minRadius môže spôsobiť, že ako kruhy budú detegované malé nečistoty v obraze. Príliš vysoká hodnota minRadius spôsobí, že vyleptané stopy , ktoré majú menší polomer nebudú detegované. Príliš nízka hodnota maxRadius spôsobí, že stopy s väčším polomerom nebudú detegované. Príliš vysoká hodnota spôsobí detegovanie kruhov, ktoré sú vpísané do oblasti medzi stopami alebo opísané okolo väčšieho počtu stôp, aj ak sa tam žiadna stopa v skutočnosti nenachádza.

4 NAVRHOVANÉ RIEŠENIE PROBLÉMU

Pre zlepšenie detekcie stôp bude vytvorený nový softvér v jazyku Java, ktorý bude automatizovať spracovanie obrazu a generovanie výstupného protokolu. V softvéri bude použitý zvolený algoritmus Houghovej transformácie na detekciu stôp. Vďaka tomu, že deteguje iba okrúhle objekty bude zmenšená pravdepodobnosť, že bude započítaný škrabanec na skle, ktorý má skôr ostré hrany.

Spolu so softvérom bude súčasťou riešenia upravený skript v makrojazyku LAS, ktorý umožní laborantovi snímkovať dozimeter jako doteraz iba bez následného počítania stôp starou metódou. Tento skript bude ukladať nasnímkované obrázky do určeného adresára tak, ako pôvodný skript.

Aby softvér lepšie automatizoval meranie a aby nebola porušená následnosť vo vyhodnocovaní vyleptaných dozimetrov a aby nedošlo ku zámene meraných dozimetrov, softvér bude použitý tesne po tom, čo laborant dokončí snímkovanie dozimetra pomocou mikroskopu.

4.1 Postupnosť vyhodnocovania dozimetra

Práca pri vyhodnocovaní dozimetra prebieha podľa nasledujúcej postupnosti úkonov:

Prvá časť – snímkovanie dozimetru:

- Vloženie dozimetrického sklíčka pod mikroskop.
- Užívateľ nastaví ľavý horný okraj na sklíčku, od ktorého bude snímkovaná meraná plocha.
- Spustenie upraveného skriptu v prostredí LAS.
- Vyplnenie názvu merania – použije sa ako názov podadresára so súbormi merania.
- Ručné zaostrenie 64 snímaných políčok pomocou špeciálnej ovládacej periférie pripojenej k mikroskopu.
- Prebieha snímkovanie políčok a ich ukladanie na HDD.

Druhá časť – vyhodnotenie snímok novým algoritmom:

- Adresár s obrázkami sa otvorí vo vytvorenom softvéri.
- Zobrazenie snímok a ich možné prehliadanie, tlačítko next a ďalšie klávesy umožnia presun na ďalšie políčko.
- Pri zobrazení novej snímky sa spustí detekecia stôp v obraze, detekcia je robená novým algoritmom.

- Je možné prehliadanie výslednej detekcie, a jej korekcia. Po vykonaní korekcie detekcie sa zmení počet detegovaných stôp a nová hodnota bude použitá v reporte. Tlačítkom next sa zobrazí ďalšie políčko.
- Vygenerovanie reportu - po spracovaní posledného políčka je možné stlačiť tlačítko uložit. Tiež je možné vypniť základné identifikačné údaje dozimetra do protokolu ako je jeho identifikácia a názov organizácie.
- Report sa ukladá na HDD do podadresára v adresári s meraním, spolu so značkou dátumu v názve adresára. V prípade opakovaného generovania reportu sa budú všetky reporty nachádzať v adresári merania, iba s iným dátumom v názve adresára reportu.

4.2 Navrhovaný algoritmus

Algoritmus, ktorý je navrhnutý na použitie používa ako najdôležitejší krok Houghovu transformáciu, ktorou sa majú detegovať vyleptané stopy na dozimetrickom sklíčku. Najprv prebehne prespracovanie obrazu jeho rozmazaním, čo je potrebné pre dobrú detekciu kruhov v obraze a nízku citlivosť na defekty v obraze.

Následne sa spustí Houghova transformácia obrazu. Výsledkom sú pozície a polomery nájdených kruhov. Jeden kruh by mal zodpovedať jednej stope, spočítaním detegovaných kruhov by sa mal hypoteticky určiť počet stôp spôsobených časticami. Niektoré kruhy sa však prekrývajú alebo sa nachádzajú menšie kruhy vnútri väčších kruhov. Tieto kruhy nadhodnocujú počet nájdených stôp a preto musia byť filtrované algoritmom hľadajúcim prekrývanie sa kruhov.

Získaný počet prefiltrovaných kruhov je výsledný odhad počtu stôp nachádzajúcich sa v obraze pochádzajúci z metódy.

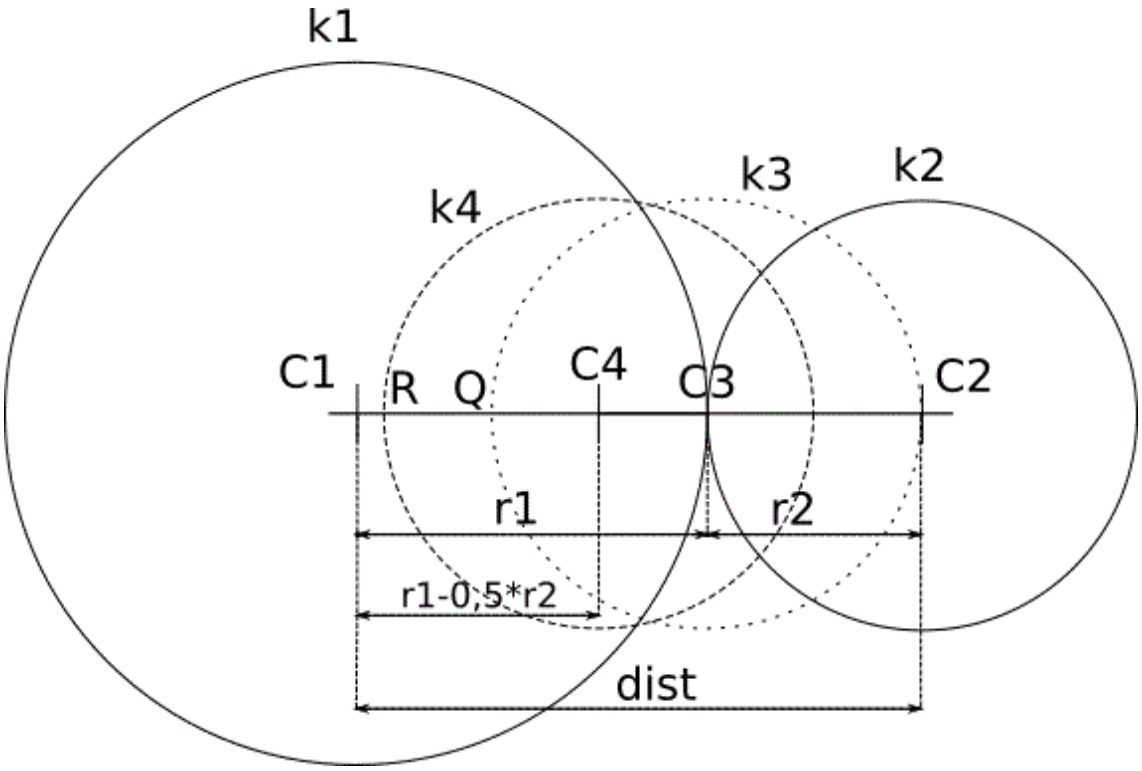
Navrhovaná postupnosť krokov pre spočítanie stôp:

- Prečítanie obrazu zo súboru
- Rozmazanie obrazu (funkcia *cv2.medianBlur*)
- Konverzia snímky na odtiene sivej (funkcia *cv2.cvtColor*)
- Houghova transformácia (funkcia *cv2.HoughCircles*)
- Získaný je zoznam nájdených kruhov obsahujúci stred kruhu a polomer.
- Vyradenie kruhov vpísaných do iných kruhov a vyradenie kruhov, ktoré sa príliš prekrývajú. (bude popísané v následujúcej podkapitole)
- Nájdenie a spočítanie reťazových stôp, odstránenie príslušných bublín zo zoznamu kruhov – tento krok bude možno pridaný v budúcnosti.
- Počet detegovaných stôp je počet kruhov, ktoré neboli vyradené zo zoznamu + počet nájdených reťazových stôp.

4.3 Filtrovanie prekrývajúcich sa kruhov

V tejto podkapitole bude popísaný algoritmus, ktorý rieši problém prekrývajúcich sa kruhov a zlepšuje výstupný zoznam stôp z Houghovej transformácie. Bez jeho použitia je metóda zaťažená vysokou chybou merania, ktorá môže dosahovať niekoľkonásobok skutočnej hodnoty počtu stôp.

Najprv je definovaná metrika, na základe ktorej sa určí, či sa 2 kruhy prekrývajú.



Obrázok 4.1 Metrika prekrývajúcich sa kruhov

Na obrázku sú vidieť 3 situácie prekrývania sa dvoch kruhov. K1 predstavuje väčší kruh. Menší kruh je predstavovaný kruhmi k2, k3 a k4, ktoré majú rovnaké polomery. Ak sa kruhy neprekrývajú a iba sa dotýkajú, potom vzdialenosť ich stredov C1 a C2 je rovná súčtu ich polomerov r1 a r2, čo zobrazuje dvojica k1 a k2. Ak sa vzdialenosť zmenší, kruhy sa začnú prekrývať. Dvojica k1 a k3 zobrazuje situáciu, kde sa stred kruhu k3 dotýka obvodu väčšieho kruhu k1 v bode C3. Menší kruh sa tu priblížil oproti predchádzajúcemu prípadu o vzdialenosť r2.

Miera prekrývania bude chápaná ako percento priemeru menšieho kruhu, ktoré leží vnútri väčšieho kruhu. V prípade k1 a k3 to je 50%, pretože na spojnici stredov kruhov vidieť, že vzdialenosť Q-C3 od priesečníku Q medzi kruhom k3 a spojnicou C1-C2 po stred C3 je rovná polomeru r3, čo je 50% priemeru kruhu k3. Vzdialenosť stredov C1-C3 je teda rovná polomeru väčšieho kruhu r1.

Tretí prípad ilustruje situáciu, kde je prekryv 75% medzi k1 a k4. Vnútri k1 leží časť spojnice stredov od bodu R po bod C3. Táto vzdialenosť je rovná 1,5 násobku polomeru r4, čo je 75% priemeru k4. Zistenie, či je prekryv viac ako 75% je použité na zistenie duplicitných detekcií kruhu, ktoré ležia veľmi blizko seba a všetky sa prekrývajú s veľkým kruhom, ktorý značí nájdenú stopu. Podmienka je definovaná následovne:

$$dist < r1 - \frac{r_2}{2}; r1 > r2$$

Kde: dist je vzdialenosť stredov kruhov,
r1 je polomer väčšieho kruhu,
r2 je polomer menšieho kruhu

Ak je podmienka splnená, kruhy sa príliš prekrývajú na to aby predstavovali 2 samostatné stopy a preto je menší kruh eliminovaný zo zoznamu stôp.

Kompletný algoritmus v pseudokóde je:
N je počet kruhov

Pre všetky nájdené kruhy I od 0 po N-1, ktoré ešte neboli vyradené:
    Pre všetky nájdené kruhy J s indexom vyšším ako I, ktoré ešte neboli vyradené:
        r1 je polomer kruhu I
        r2 je polomer kruhu J
        dist je vzdialenosť kruhov I, J
        ak je r1 > r2 a ak kruh:
            ak je dist < (r1 – 0,5 * r2):
                potom kruh J je vyradený zo zoznamu
        ak je r1 ≤ r2:
            ak je dist < (r2 – 0,5 * r1):
                potom kruh I je vyradený zo zoznamu

Z algoritmu je vidieť, že vždy je vyraďovaný ten menší kruh, pokiaľ jeho prekryv s väčším kruhom je viac ako 1,5 násobok polomeru menšieho kruhu. Postupne sa kruhy v zozname vyraďujú a ďalej neovplyvňujú výsledok.

4.4 Chyby pri detekcii

Následuje kompletnejší zoznam možných chýb pri detekcii, ktoré tiež nadhodnocujú výslednú zmeranú hodnotu a tiež sú uvedené možnosti, ako sa dajú nadbytočné detekcie eliminovať prácou s parametrami Houghovej transformácie alebo úpravou celkového algoritmu:

- Príliš malý kruh, môže ísť o kaz na povrchu polyméru. Problém sa dá riešiť určením minimálneho priemeru kruhu, ktorý sa akceptuje pri detekcii.
- Príliš veľký kruh – algoritmus má tendenciu opisovať veľkú kružnicu okolo väčšieho zhluku kruhov spôsobených naleptanými stopami, ak spolu pripomínajú kruh. Je to z toho dôvodu, že algoritmus nerozoznáva jednotlivé objekty v obraze ale miesto toho hľadá miesta, kde sa môže vizuálne nachádzať kruh. Tento problém sa vyrieši stanovaním maximálneho polomeru kruhu, ktorý sa bude akceptovať pri detekcii. Hodnota tohto rozmeru sa určí tak, aby bola väčšia ako väčšina typicky sa vyskytujúcich rozmerov stôp na dozimetroch.
- Detegovanie kruhu vpísaného vo väčšom kruhu, môže ísť o falošnú detekciu. Tento problém rieši spomínaný algoritmus na nájdenie prekrývajúcich sa kruhov.
- Detegovanie "chvostu" – ide o stopy, ktoré majú ďalšiu menšiu bublinu vidieť na boku väčšej bubliny, spravidla s viac ako 50% prekryvom priemeru menšej bubliny. Pravdepodobne ide o pokračovanie stopy vo väčšej hĺbke polyméru. Tieto stopy sa na snímkach vyskytujú dosť často. Tento typ chyby súvisí s interpretáciou významu kruhu v obraze. Stopa vznikla šikmým preletom alfa častice cez materiál a je naznačená najprv väčšou guličkou a pokračuje guličkou menšou. Ak sa zarátajú oba kruhy, potom hrozí, že sa budú stopy zarátavať dvojnásobne. Tento problém rieši algoritmus pre hľadanie prekrývajúcich sa kruhov.
- Kruhy, ktoré sa príliš prelínajú, môže ísť o falošné stopy. Tento problém tiež rieši algoritmus pre hľadanie prekrývajúcich sa kruhov.
- Reťazové stopy, ktoré sú popísané v zvlášť kapitole.

4.5 Reťazové stopy

Ďalšou uvažovanou možnosťou na obmedzenie falošných detekcií je detegovanie reťazových stôp, ktoré sú tvorené väčším počtom bublín, ktoré sú vedľa seba zoradené približne do priamky, ktorá je ale krivá a s dosť rôznorodými prekryvmi susedných bublín. Príklad reťazovej stopy je na obrázku v prílohe A.1.

Tieto bubliny patria k jednej stope alfa častice, ktorá letela po istej dráhe cez materiál a po ceste spôsobovala molekulárne poškodenie polyméru. Avšak algoritmus ich môže označiť ako veľké množstvo samostatných stôp, čo vedie ku veľkej chybe merania.

Zatiaľ je tento problém vyriešený manuálnou korekciou hodnoty laborantom v softvéri. V ďalších verziách bude mať možnosť detekciu ručne opraviť vylúčením týchto stôp z detekcie pomocou nakreslenia polygónovej oblasti v obraze.

Alternatívou by mohla byť automatická detekcia reťazových stôp. K jednému z kritérií na ich detekciu je, že ich uvažované stredy musia byť všetky dostatočne priestorovo korelované na jednej lineárnej priamke. Ďalej musia byť susedné bubliny podobnej veľkosti ináč sú to nesúvisiace bubliny, ak susedia, tak sa totiž nachádzajú v podobnej hĺbke obrazu. Musí byť empiricky stanovený rozsah vyžadovaného prekryvu medzi susednými bublinami, čo implikuje, že bubliny nesmú byť ďaleko od seba a tak teda stopa musí byť súvislá. Tiež treba použiť limit na minimálny počet bublín v stope, napríklad 4. Boli tiež pozorované reťazové stopy s dvoma a viacerými radmi bublín a tiež stopy, ktoré nemali príliš lineárnu dráhu ale skôr krivku.

Ďalej, väčšia časť stôp rovnobežných s povrchom je fotografovaných ďalej od povrchu, z dôvodu, že tesne pri povrchu je leptaním odobraná časť polyméru. Z toho dôvodú sú bubliny menšieho priemeru ako najväčšie bubliny fotografované pri povrchu.

5 LADENIE PARAMETROV HOUGHOVEJ TRANSFORMÁCIE

V následujúcej kapitole sú popísané výsledky rôznych pokusov, ktoré boli robené s ukážkovým kódom pre detekciu kruhov z dokumentácie knižnice OpenCV[28]. Je uvedený vplyv zmeny na detekciu a dôsledky na presnosť detekcie ak by bola zmena použitá. Boli identifikované optimálne parametre Houghovej transformácie, ktoré sa ďalej použijú pri testovaní na dátovej sade a v hotovej grafickej aplikácii.

5.1 Testovací kód

Testovací skript je napísaný v jazyku python a je kompatibilný s interpreterom python 2.7.11. Verzia knižnice OpenCV2 je 3.0.0 a knižnica numpy má verziu 1.11.1. Skript je kompatibilný s operačným systémom Windows 10 Home 64 bit. Testovací skript je priložený v prílohe B.

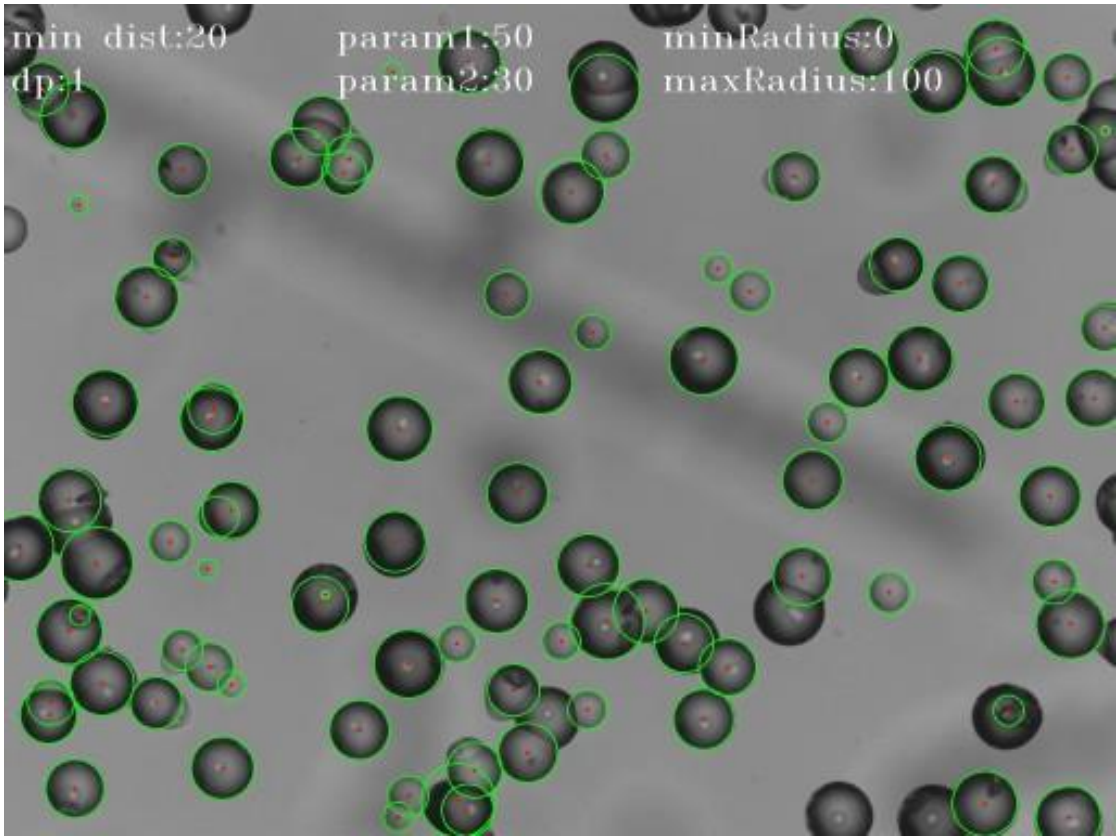
V kóde je metóda *run* automatizujúca experiment so zadanými parametrami transformáciami. najprv načíta testovací obrázok v odtieňoch sivej farby. Ako prvý krok sa obrázok rozmaže pomocou funkcie *GaussianBlur*. Veľkosť rozmazávacej matice je nastavená pôvodne na 9x9 parametrom *ksize*. Následne sa metódou *houghCircles* detegujú kruhy. [27] Nájdené kruhy sú dokreslené zelenou farbou do obrázku spolu s textovým popisom použitých parametrov bielou farbou. Kruhy niesú filtrované algoritmom na elimináciu prekrývajúcich sa kruhov.

Postupne bola volaná metóda run aby boli rozmietané jednotlivé parametre od počiatočných ručne určených parametrov:

blur=9, dp=1, mindist=20, param1=50, param2=30, minRadius=0, maxRadius=100

Pre tieto počiatočné parametre sú vyznačené stopy vidieť na následujúcom obrázku. (Obrázok 5.1) V pozadí je rozmazaná veľká reťazová stopa spolu s ďalšími rozmazanými guličkami. Stopy pochádzajú z dozimetra, ktorý dostal dosť veľkú dávku žiarenia. Preto počet stôp je veľký a preto dochádza s vyššou pravdepodobnosťou ku prekryvu stopy. Detekcia fungovala pre túto snímku spoľahlivo, nevyznačené sú hlavne stopy na okraji snímky. V strede snímky sa nachádza zopár nevyznačených stôp, ktoré sú prekryté inými stopami. Určenie polomeru a stredu kruhu je pri týchto parametroch pomerne presné. Niektoré stopy však boli označené menším kruhom, pretože sa javia ako vnútorný svetlejší kruh. Zopár stôp má na tejto snímke chvost, no tie boli nevýrazné a možno preto neboli vyznačené. V spodnej časti snímky vidieť jeden veľký kruh, ktorý opísal 2 stopy a pretína 2 daľšie stopy. Takýmto chybným detekciám je ťažké zabrániť.

Pôvodný nevyznačený testovací obrázok spolu s ďalšími príkladmi výsledných detekcií je v prílohe C.



**Obrázok 5.1 Detekcia pomocou počiatočne zvolených parametrov.
Niektoré stopy nie sú detegované, avšak množstvo falošných detekcií je minimálne.**

5.2 Efekt rozmazania

Ide o prvý krok predspracovania obrazu pred samotnou detekciou kruhov. Jeho úplné vynechanie spôsobilo nárast falošných detekcií kruhov napríklad na pozadí, ktoré obsahovalo iba rozmazané stopy hlbšie v materiáli, ktoré boli akoby na pozadí obrazu. Ďalšie kruhy boli opísané okolo viacerých kruhov alebo vpísané do medzier medzi kruhmi zoskupenými približne do polkruhu a podobne.

5.3 Parameter dp

Parameter určuje pomer veľkosti akumulátorovej matice, ktorá sa používa pri hľadaní kruhov pomocou Houghovej transformácie. Pokusmi bolo zistené, že nastavenie hodnoty 1 vedie ku pomerne normálnym výsledkom a tiež sú presne určované stredy kruhov. Avšak nastavenie hodnôt od 2 až po 5 viedlo ku veĺkému zvýšeniu počtu falošných detekcií, ktoré graficky zaplnili celý obrázok a navyše sa spomalil algoritmus.

Postupne sa pre hodnoty dp od 10 po 40 zmenšuje počet falošných detekcií až na minimálne hodnoty pri hodnotách 45 a vyššie. Avšak, algoritmus prestáva merať presne priemer u tých bublín, ktoré vedel zmerať pomerne presne a takisto dosť klesá schopnosť

presne zamerať stred kruhu. Od hodnôt 75 a vyššie sa stráca schopnosť detegovať niektoré bubliny, čím vyššie hodnota je, tým menej ich je nájdených.

Nepresné meranie stredu a priemeru kruhov pri vyšších hodnotách dp by znemožnilo detekciu ‘chvostu’, resp. pridruženej malej bubliny ku veľkej bubline s ktorou sa prekrýva. Tie patria k jednej stope.

Testovacie podmienky boli:

blur=9,mindist=20, param1=50, param2=30, minRadius=0, maxRadius=100.

5.4 Minimálna vzdialenosť medzi kruhmi

Detekcia je citlivá na zmenu parametra minimálnej vzdialenosti medzi dvoma detegovanými kruhmi. Pri príliš nízkej hodnote v rádovo jednotiek pixelov prudko narastie počet detegovaných kruhov, ktoré sa všetky koncentrujú na mieste kde sa skutočne nachádza kruh, jeden kruh je nájdený viacnásobne. Takisto sa spomaľuje beh detekcie. Pri hodnotách minDist 15 až 20 sa odstráni väčšina viacnásobne detegovaných stôp bez straty detekcie niektorých stôp. Pri nižších hodnotách narastal počet viacnásobných detekcií.

Testovacie podmienky boli:

blur=9, param1=50, param2=30, minRadius=0, maxRadius=100, dp=1.

5.5 Param1

Pri nízkych hodnotách tohto parametru je detekcia pomalá, napríklad hodnota 10 je ešte stále pomalá, hodnoty 20 a vyššie sú dostatočne rýchle. So stúpajúcou hodnotou klesá počet detegovaných kruhov. Hodnota 30 mala nízky počet nadbytočných detekcií a ešte nestrácala niektoré stopy v obraze. Hodnota 35 mala menší počet viacnásobných detekcií ale začali sa strácať predtým detegované stopy.

Testovacie podmienky boli:

blur=9,mindist=20, param2=30, minRadius=0, maxRadius=100, dp=1.

5.6 Param2

Ak je tento parameter nastavený na nízku hodnotu, potom je snímka pokrytá veľkým počtom falošne detegovaných kruhov a taktiež je detekcia pomalá. Pri hodnote 25 klesol počet falošných detekcií na únosnú úroveň, avšak niektoré stopy majú vpísané malé kruhy. Pri hodnotách od 30 a viac prestali byť detegované niektoré stopy.

Testovacie podmienky boli:

blur=9, mindist=20, param1=50, minRadius=0, maxRadius=100, dp=1.

5.7 Minimálny polomer kruhu

Parameter nemal moc veľký vplyv na počet falošných detekcií ani na rýchlosť algoritmu. Pri hodnote 15 a vyššej sa prestali detegovať najmenšie stopy aké boli na snímke. Pri hodnote 35 sa prestali detegovať dostatočne veľké stopy, ktoré sú potrebné pre správne meranie. Praktická hodnota pre výsledný algoritmus by mohla byť 10, aby sa zachovala detekcia aj najmenších stôp.

Testovacie podmienky boli:

blur=9,mindist=20, param1=50, param2=30, maxRadius=100, dp=1.

5.8 Maximálny polomer kruhu

Typická stopa nemá na snímke robenej s objektívom 10x väčší priemer ako približne 140 pixelov, čo dáva maximálny polomer 70 pixelov. Tento rozmer bol zistený prezeraním snímok zo slovenských jaskýň. S istou rezervou 20% pre stopy s viac elipsoidným tvarom sa dá určiť maximálny polomer kruhu 85 pixelov.

5.9 Skombinovanie doposiaľ nájdených parametrov

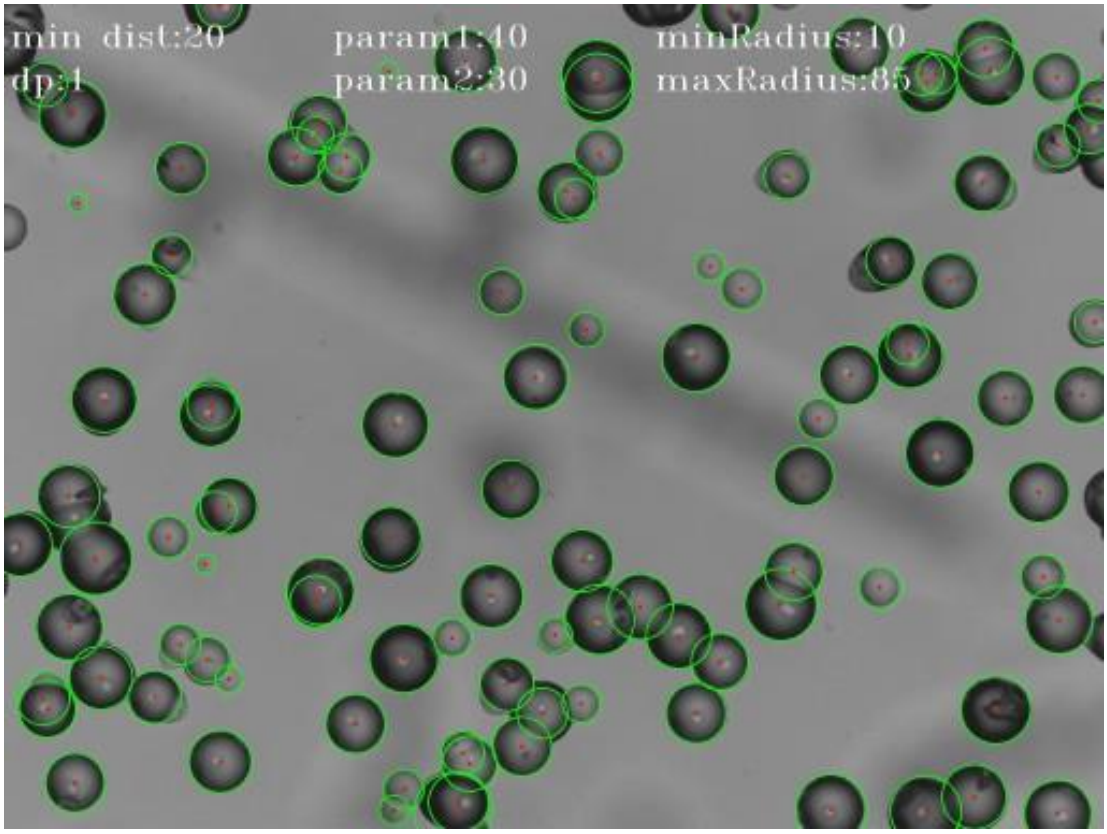
Následujú parametre, ktoré boli identifikované v predchádzajúcich podkapitolách:

blur=9, dp=1, mindist=20, param1=30, param2=25, minRadius=10, maxRadius=85

Pri použití týchto parametrov vyzerá výsledok ako na obrázku v prílohe C.3. Aj stopy v pravom dolnom rohu, ktoré presahujú cez okraj sa detegujú, aj keď sa pri vyšších hodnotách param1 a param2 nedetegovali. Avšak obraz obsahuje plno prekrývajúcich sa a vpísaných kruhov. Ak by sa malo toto riešenie použiť, musel by sa pridať ďalší krok detekcie, a to odstránenie kruhov, ktoré sú vpísané do väčšieho kruhu a kruhov ktoré sa prekrývajú nad určitú mieru s iným kruhom.

5.10 Zníženie počtu vpísaných kruhov na úkor citlivosti detekcie

Param1 bol oproti parametrom z podkapitoly 5.9 zvýšený na hodnotu 40 a param2 zvýšený na hodnotu 30. Dosiahol sa určitý kompromis, pri ktorom sa nestráca detekcia skoro žiadnej stopy ale poklesne počet vpísaných kruhov a falošných detekcií na minimum. Avšak tie sa stále budú musieť filtrovať podľa popisu v kap. 5.9. Tento postup nestráca detekciu väčšiny stôp, ktoré sú na okraji snímky.



Obrázok 5.2 Detekcia s parametrami so zvýšenou hodnotou param1 a param2.

5.11 Detegovanie chvostu

Pri experimentovaní boli spozorované situácie, keď veľké stopy mali malú pridruženú bublinu, ale veľká stopa bola detegovaná ako menšia, a z pohľadu detekcie sa javila byť vzdialená a oddelená od malej bubliny, čo v týchto prípadoch znemožňuje vyradiť menšiu bublinu ako spôsobenú jednou stopou. Houghova transformácia má problémy rozpoznať správne priemer niektorých kruhov.

5.12 Výsledky ladenia parametrov

V tejto kapitole bolo experimentované s parametrami Houghovej transformácie, ktorá detegovala kruhy v obraze. Metóda je implementovaná v knižnici OpenCV 2, a testovací skript bol napísaný v jazyku python.

Pred samotnou detekciou kruhov bolo potrebné obraz rozmazať, napríklad pomocou mediánového filtra, v OpenCV funkciou *medianBlur*. Bez tohto kroku narastá množstvo falošných detekcií kruhov, takže tento krok sa nedá vynechať.

Menené parametre boli dp, minimálna vzdialenosť medzi kruhmi, parameter1, parameter2, minimálny polomer detegovaného kruhu, maximálny polomer kruhu a bol zhodnotený vplyv zmeny parametrov.

Vstupný testovací obrázok sa dá nájst v prílohe C aj spolu s príkladmi detekcie pri rôznych parametroch.

Parameter dp nastavený na hodnotu 1 ponúka presné meranie stredu stopy a pri väčšom množstve stôp je správne zmeraný priemer kruhu, ktorý sa kryje s obrysom stopy, avšak tiež s istou malou odchýlkou. Hodnoty 2 až 40 majú vysoký počet falošných detekcií kruhov. Od hodnôt od 40 po 70 má meranie podobnú kvalitu ako pri hodnote 1 ale s nižšou presnosťou určenia stredu a priemeru bubliny, čo by znemožnilo ďalšie filtrovanie bublín, ktoré sú pridružené k rovnakej stope, ktoré by nadhodnocovali počet stôp. Hodnoty nad 75 spôsobia postupný nárast nedetegovaných stôp v obraze, ktorý by spôsobil podhodnotenie počtu napočítaných stôp.

Optimálna hodnota minimálnej vzialenosti stredov kruhov bola medzi 15 až 20 pixelov. Zvyšovaním tohto parametru klesal počet viacnásobných detekcí, nad optimálnou hodnotou prestali byť detegované niektoré stopy.

Param1 a param2 mali veľký vplyv na rýchlosť detekcie a takisto na počet falošných detekcií. Ich príliš nízka hodnota spôsobovala pomalý beh algoritmu a pri príliš vysokej hodnote prestali byť detegované stopy. Optimálna hodnota param1 bola Približne 30, pre param2 bola 25.

Optimálna hodnota pre minimálny polomer kruhu je 10 ak je snaha detegovať aj najmenšie stopy, ktoré niesú moc kontrastné a môžu sa stratiť v defektoch v obraze. Ak sa veľmi malé stopy nebudú detegovať pre zvýšenie odolnosti voči falošným detekciám, potom bude optimálna hodnota 35.

Maximálny polomer kruhu môže byť nastavený na 85 pixelov, do ktorého sa mestí väčšina stôp v testovacích dátach.

Celkovo bolo pozorované spomalenie algoritmu, ak dochádzalo k veľkému počtu detekcií kruhov, zväčša falošných detekcií.

Optimálne parametre, ktoré nestrácajú stopy, fungovali na testovací obrázok, ktorý je v prílohe C, avšak detekcia produkuje vpísané kruhy do väčšieho kruhu vyznačujúceho detegovanú stopu. Pre vyfiltrovanie výstupu bude použitý algoritmus, ktorý nájde a odstráni kruhy vpísané do väčšieho kruhu.

Alternatívou by bolo použiť mierne vyššie hodnoty minDist, param1 ,param2 alebo minRadius, aby sa vpísané kružnice nevytvárali za cenu toho, že niektoré stopy nebudú detegované. Takisto by sa zvýšila odolnosť voči falošným detekciám. Správny výsledok bude napriek chýbajúcim stopám zaistený kalibráciou meracieho systému.

S určenými parametrami pre detekciu bude ďalším krokom porovnanie detekcie s pôvodným algoritmom, či došlo k zlepšeniu detekcie a testovanie filtrácie kruhov.

6 POROVNANIE ALGORITMOV NA POČÍTANIE STÔP

V tejto kapitole bude navrhovaný algoritmus používajúci Houghovu transformáciu, ktorý je vylepšený o filtráciu prekrývajúcich sa kruhov, spustený na sade snímok 116 dozimetrov použitých na sledovanie radiačnej záťaže jaskyniarov v slovenských jaskyniach. Na základe počtu nájdených stôp na dozimetri bude porovnaná nová detekcia s pôvodnou. Bude zhodnotená kvalita detekcie starej aj novej metódy.

6.1 Dátová sada

Radón sa meral v 10 jaskyniach na Slovensku počas 3. kvartálu roku 2015. Dozimetre boli nosené pracovníkmi a niektoré dozimetre boli ponechané v kancelárii správcu jaskyne, kde merali pozaďovú hodnotu žiarenia danej lokality. Pozaďové dozimetre absorbujú malú dávku žiarenia a preto majú malé počty stôp. Počty meraní v jednotlivých jaskyniach boli rôzne, od 7 po 22, v súčte je dostupných 116 dozimetrov vyhodnotených pomocou starej metódy. K dispozícii sú pôvodné snímky dozimetra z mikroskopu pri zväčšení 10x. Každý dozimeter má 64 snímok segmentov, preto je v dátovej sade 7424 obrázkov. Dostupné sú počty stôp nájdených na jednotlivých segmentoch pomocou algoritmu Watershed a taktiež celkový súčet počtu stôp. Avšak niesú dostupné pozície a veľkosť nájdených stôp, ktoré by uľahčili porovnávanie metód. Pôvodné makro z prostredia LAS tieto údaje neukladá.

6.2 Testovací program

Program na spracovanie celej dátovej sady dozimetrov bol prepísaný do jazyka Java a zároveň bola v tom momente začatá implementácia modelovej vrstvy vznikajúcej grafickej aplikácie. Boli vytvorené triedy, ktoré ukladajú dáta v xml a csv formáte pre jednoduché exportovanie dát pre štatistické spracovanie. Viacej o architektúre aplikácie bude v kapitole o grafickej aplikácii.

Zvolené parametre Houghovej transformácie boli tie, pri ktorých je redukovaný počet vpísaných kruhov vo väčších kruhoch a tiež počet falošných detekcií:

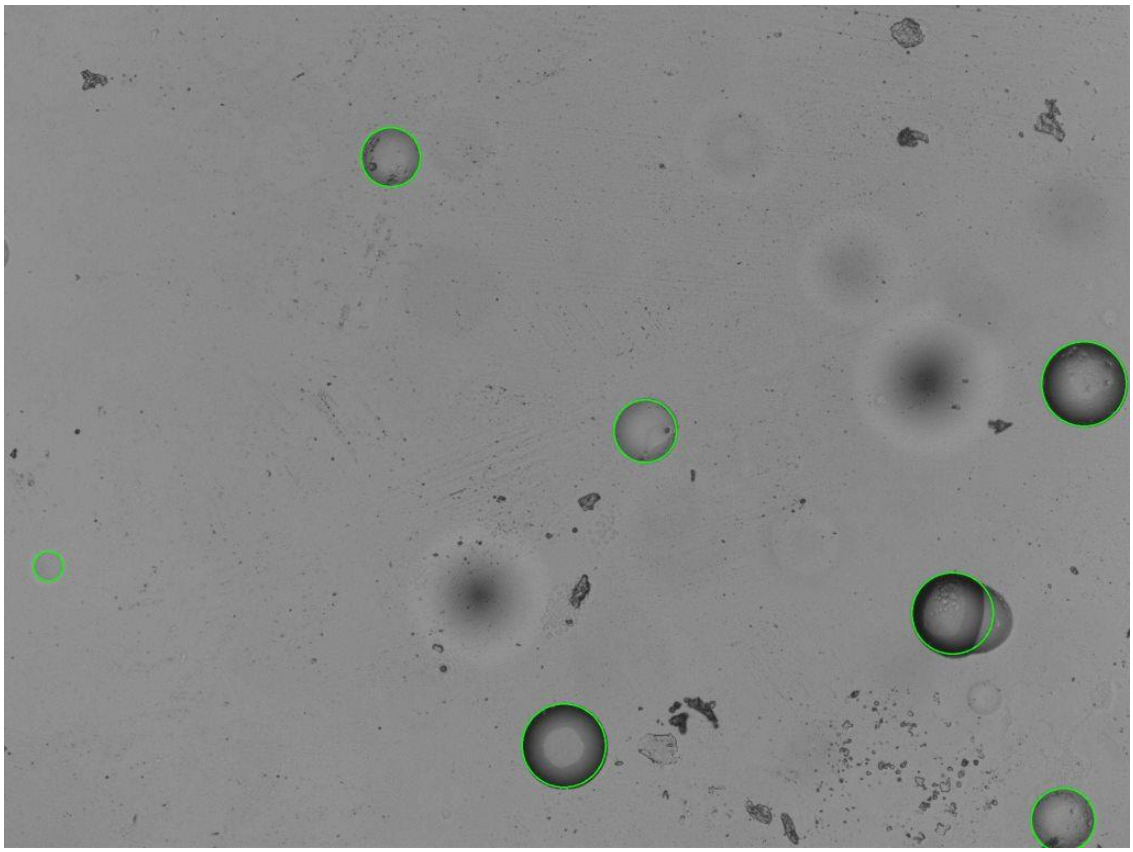
blur=9, dp=1, mindist=20, param1=40, param2=30,
minRadius=10, maxRadius=85

Houghova transformácia vyprodukuje zoznam kruhov v obraze a tie sa už filtrujú algoritmom na vyradenie prekrývajúcich sa kruhov, narozdiel od predchádzajúceho programu na ladenie parametrov. Program postupne vyhodnotil každý dozimeter a spočítal počet stôp na jednom dozimetri a počet stôp na jeho 64 jednotlivých segmentoch.

Program bežal 4958 sekúnd, čo je necelých 83 minút na procesore i5 4210U 1.7GHz na jednom vlákne. Pri každom segmente došlo ku načítani obrázku z disku, detekcia jeho kruhov volaním houghCircles z knižnice OpenCV [27], vyznačenie kruhov na obrázku a nakoniec uloženie výsledného obrázku na disk. Z toho sa dá spočítať, že na jeden segment pripadá 667 milisekúnd behu programu. S podobnou dobou treba počítať aj pri detekcii jedného segmentu vo vznikajúcej grafickej aplikácii.

6.3 Obrazové výstupy

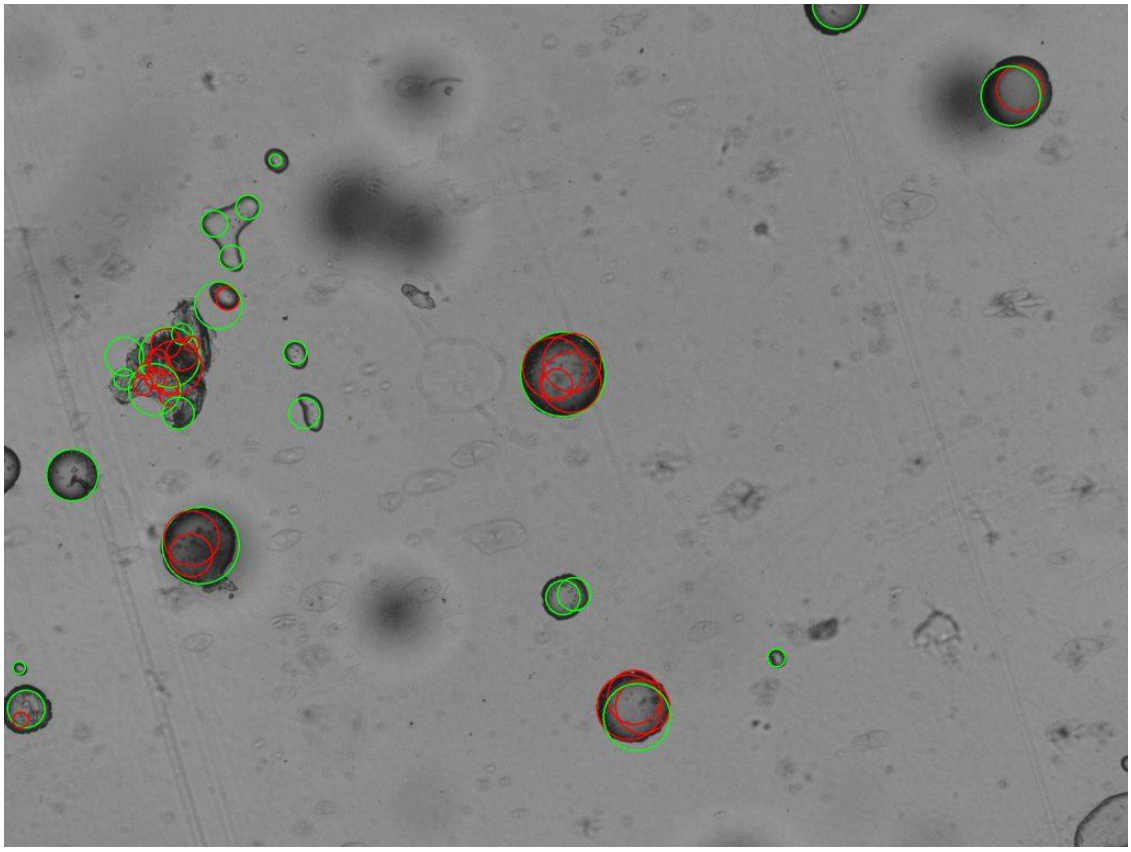
Algoritmus vyznačuje kruhové stopy alfa častíc v obraze. Vo výstupnych obrázkoch sú prijaté stopy vyznačené zelenou farbou. Tie sa započítajú do výsledného počtu stôp. Zamietnuté kruhy sú vyznačené červenou farbou.



Obrázok 6.1 Správne vyznačená snímka.
Snímka pochádza z dozimetra AE2 z Belianskej jaskyne. Malé defekty na povrchu niesú vyznačované algoritmom.

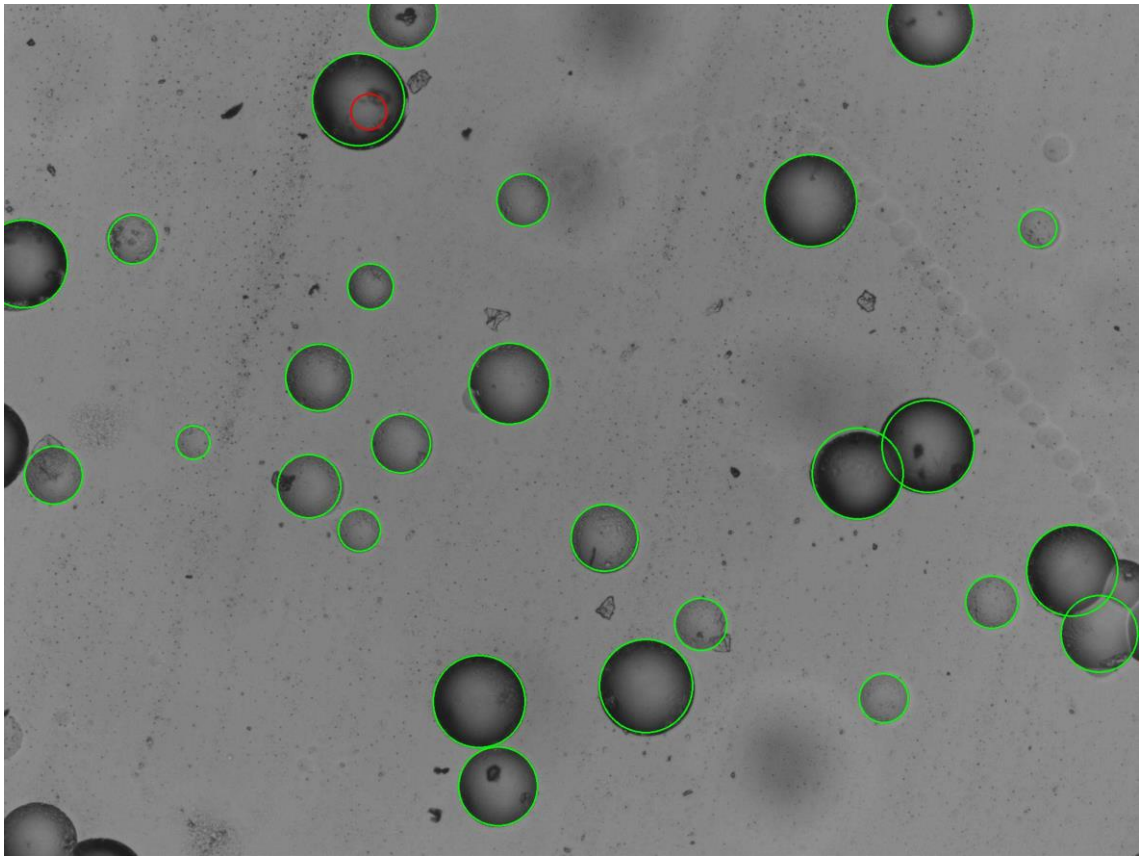
Na prvom obrázku (Obrázok 6.1) vidieť príklad detekcie, pri ktorej nedošlo ku vyznačeniu falošných stôp. V pravej dolnej časti je vidieť stopa, ktorá má “chvost”, ktorý ale nebol vyznačený. Ak sa chvost nachádza dosť ďaleko od pôvodnej bubliny, potom býva chybne vyznačený.

Na ďalšom obrázku (Obrázok 6.2) vidieť ako boli v ľavom hornom rohu označené defekty v obraze ako vyleptané plochy na povrchu dozimetra. Tiež sú defekty plné prekrývajúcich sa kruhov, ktoré sú eliminované. V strede a na spodnej časti sú 2 stopy, ktoré boli vyznačené prekrývajúcimi sa kruhmi, ktoré však boli eliminované pomocou algoritmu hľadajúceho kruhy, ktoré sa prekrývajú o viac ako 75% svojho priemeru s väčším kruhom. Algoritmus v tomto prípade fungoval správne a zmenšil počet falošných detekcí, ktoré by viedli k nadhodnoteniu počtu stôp jednou stopou. Avšak sa na snímke nachádza jedna malá stopa, kde bol prekryv menší a tam boli obe detekcie ponechané.

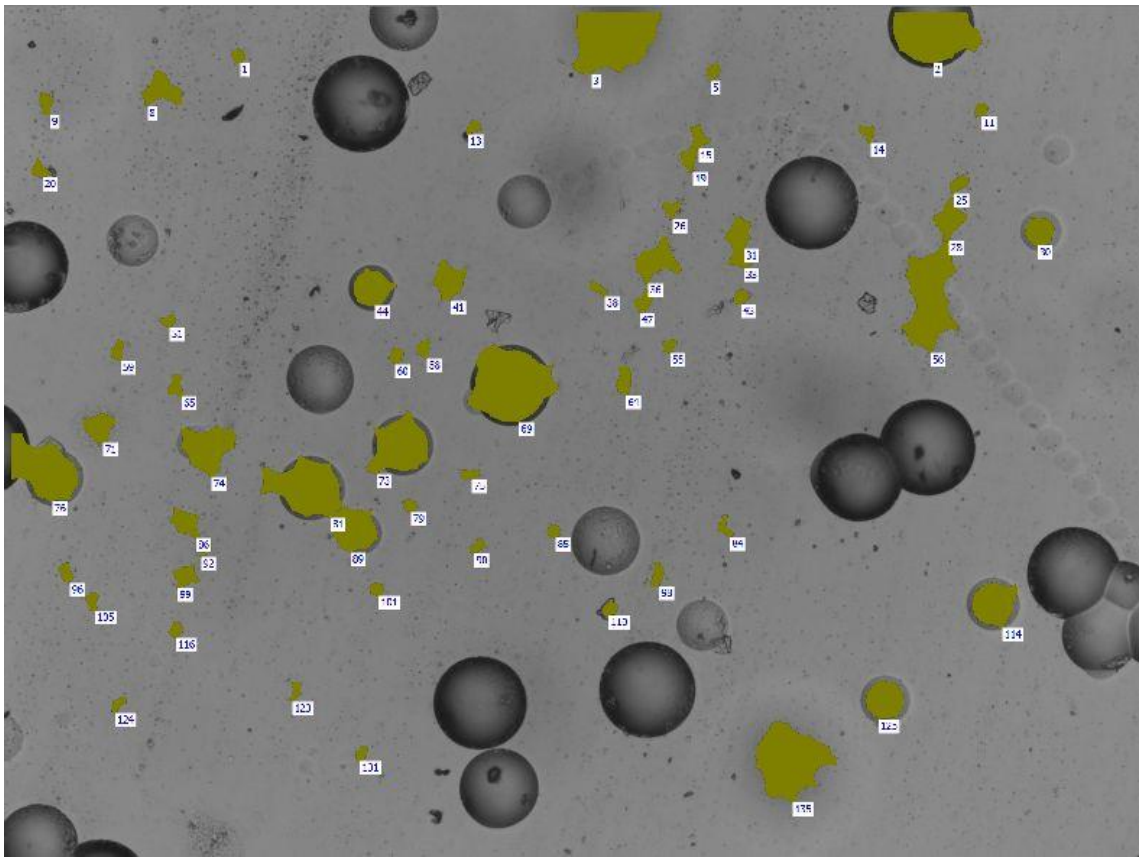


Obrázok 6.2 Snímka dozimetru AE2 segmentu 23.
Vľavo hore vidieť nesprávne označený lokálny defekt na povrchu dozimetru. Zeleným sú algoritmom vyznačené stopy, červeným sú kruhy, ktoré boli filtrované algoritmom hľadajúcim prekryv.

Na ďalších dvoch obrázkoch bude ukázané kvalitatívne porovnanie detekcie rovnakej snímky oboma metódami. Nová metóda si poradila so snímkou, na ktorej sú malé jemné škrabance a malá nevýrazná reťazová stopa v pravej časti. (Obrázok 6.3) Vyznačila 27 stôp a nevyznačila 3 stopy presahujúce okraj snímky. Filtrovaný bol jeden vpísaný kruh. Oproti tomu stará metóda má na výstupe číslo 57 stôp, ale vyznačila iba 11 skutočných stôp, ktoré označila aj nová metóda. (Obrázok 6.4) Stará metóda mala tendenciu vyznačovať rozmazané stopy na pozadí a rôzne malé defekty v obraze.



Obrázok 6.3 Detekcia snímky pomocou novej metódy. (27 stôp)



Obrázok 6.4 Detekcia snímky pomocou starej metódy. (59 stôp)

Počet stôp, ktoré pri tejto snímke vyznačila stará metóda je 220% hodnoty novej metódy, ktorá vyznačila väčšinu stôp správne. Pre tieto prípady snímok je preto stará metóda zaťažená vyššou chybou ako nová metóda. Pri starej metóde akoby neexistovala tendencia, že defekty sú lokalizované na jednom mieste, miesto toho sú falošné detekcie po celej ploche snímky.

6.4 Porovnanie nameraných údajov

Na štatistické spracovanie údajov a na vytvorenie frekvenčného polygónu bol použitý program StatGraphics 6. Počty stôp z jednotlivých metód sú v prílohe D.

Ako prvý výpočet boli urobené priemery meraní za jednotlivé jaskyne z pôvodnej a novej metódy spolu s ich 95%-ným intervalom spoľahlivosti. Z tabuľky pre pôvodnú metódu vidieť, že interval spoľahlivosti pre spočítané priemery za jaskyňu je veľmi široký a preto tieto priemery majú horšiu výpovednú hodnotu. Jasovská jaskyňa mala najšírší interval spoľahlivosti hodnoty priemerného počtu stôp na dozimetri. Hodnota priemeru je 2861,57±291,97.

Priemery počtu stôp na dozimetri z pôvodnej metódy:

Jaskyňa	počet dozimetrov	Priemerný počet stôp	95 Percentný interval spoľahlivosti pre priemer	
			dolný	horný
Belianska	19	1242,84	1065,62	1420,06
bystrianska	11	2839,64	2606,72	3072,55
demänovska	22	1594,82	1430,12	1759,51
Domica	9	1651,56	1394,06	1909,05
Driny	9	3308,67	3051,17	3566,16
gombaseká	7	2238,57	1946,60	2530,54
harmanecká	9	1955,44	1697,95	2212,94
Jasov	7	2861,57	2569,60	3153,54
ochtinská	15	2563,20	2363,75	2762,65
Važecká	8	2670,50	2397,39	2943,61
Total	116	2135,26	2063,54	2206,98

V prípade novej metódy sú intervaly spoľahlivosti veľmi široké v porovnaní so samotnou hodnotou priemeru, viac ako v prípade pôvodnej metódy. Najširší interval spoľahlivosti majú jaskyne Gombasecká s hodnotou priemeru 1040,86±719,11 a Harmanecká s priemerom 2623,11±719,11. Preto majú tieto priemery malú výpovednú hodnotu o skupine dozimetrov, ktoré popisujú. Dôvod bude zrejmý z grafu frekvenčného polygónu a z rozptylového grafu.

Priemery počtu stôp na dozimetri z novej metódy:

jaskyňa	počet dozimetrov	Priemerný počet stôp	95 Percentný interval spoľahlivosti pre priemer	
			dolný	horný
belianska	19	851,16	414,68	1287,64
bystrianska	11	2694,27	2120,62	3267,92
demänovska	22	920,09	514,46	1325,72
domica	9	1715,78	1081,58	2349,97
Driny	9	4931,11	4296,92	5565,30
gombaseká	7	1040,86	321,75	1759,97
harmanecká	9	2623,11	1988,92	3257,30
jasov	7	2879,00	2159,89	3598,11
ochtinská	15	6550,53	6059,29	7041,78
važecká	8	3751,13	3078,46	4423,79
Total	116	2630,92	2454,27	2807,57

Na následujúcom frekvenčnom polygóne vidieť porovnanie výstupných počtov stôp pre obe metódy, z ktorého sa dá odhadnúť, ktorá metóda nájde väčší počet stôp.

Obrázok 6.5 Frekvenčný polygón ukazujúci ako často sa vyskytovali rôzne počty stôp. Pôvodná metóda spracováva väčšinu dozimetrov s väčším počtom stôp ako nová.

Nová metóda našla hodnoty približne od 500 po 2000 stôp na väčšine analyzovaných dozimetrov. Stará metóda spracovala väčšinu dozimetrov s hodnotami približne od 1000 stôp po 3000. Stará metóda teda ukazuje viacej stôp na väčšom počte dozimetrov. Dá sa to vysvetliť tým, že stará metóda vyznačuje veľké množstvo falošných

stôp aj na segmentoch s pomerne kvalitným povrchom. Oproti tomu nová metóda vyznačuje stopy na lokalizovaných defektoch na niektorých segmentoch. Tiež sa vyskytujú extrémne prípady keď je celá plocha segmentu vyznačená veľkým množstvom falošných stôp.

Výsledky z novej metódy niesú rozložené tak kompaktne ako zo starej metódy, čo môže byť vysvetlené tým, že v dátovej sade sa vyskytovali aj silno ožiarené dozimetre, ktoré vytvárajú dlhý chvost distribúcie na grafe doprava. Taktiež sa dá nerovnomerná distribúcia vysvetliť tými dozimetrami, kde bolo veľké množstvo falošných stôp, ktoré vytvorili odľahlé hodnoty.

Z frekvenčného polygónu ďalej vidieť, že je problematické spracovávať nerovnomernú distribúciu hodnôt ako normálnu distribúciu, čo spôsobí, že priemer spočítaný zo všetkých dozimetrov alebo súhrnne za dozimetre z jednej jaskyne bude mať nízku výpovednú hodnotu. Ak by sa dala na porovnanie použiť parametrická štatistika, potom by sa mohol použiť medián zo 116 výsledkov pre každú metódu. Ten nebude ovplyvnený dlhým chvostom v distribúcii.

Pre lepšie porovnanie metód boli použité metodiky Znamienkový test a Wilcoxonov test. Princípom týchto testov je, že zisťujú, či oba vzorky majú rovnaký medián. Aplikujú sa na párové hodnoty. Wilcoxonov test je senzitívnejší.

Porovnávajú sa výsledky merania dozimetrov súhrnne za jednotlivé jaskyne. Na základe interpretu sa potom dá zistiť ako sa merania v jaskyni zhodujú podľa oboch detekčných metód.

Tabuľka neparametrických testov na rozdiel v hodnotách merania pre jednotivé jaskyne:

Porovnanie meraní podľa dvoch metód v jaskyniach							
Por	Jaskyňa	Znamienkový test			Wilcoxonov test		
		Z	p	Interpret.	Z	p	interpret
1	Belianska	3,67065	0,0002	+++	3,48095	0,000499	+++
2	Bystrianska	1,80907	0,07044	NS	1,11139	0,266401	NS
3	Demänovská	4,4722	0,00000	+++	4,12314	0,00000	+++
4	Domica	0066667	0,504983	NS	0,71081	0,477194	NS
5	Driny	2,66667	0,007660	++	2,72483	0,00643	++
6	Gombasecká	1,51186	0,13057	NS	2,28192	0,02249	+
7	Harmanecká	2,66667	0,00766	++	2,72483	0,00645	++
8	Jasovská	0,0000	1,0000	NS	0,25354	0,79984	NS
9	Ochtinská	3,6147	0,00030	+++	3,43617	0,00059	+++
10	Važecká	1,06066	0,28884	NS	1,89038	0,08707	NS
	Spolu	1,02132	0,3077	NS	0,65979	0,509381	NS

Z je hodnota testovacieho kritéria

P je hladina významnosti (p-hodnota), ktorú interpretujeme spôsobom:

ak je $p>0,05$, je tento test bez významu, rozdiel nie je potvrdený, označuje sa NS (Non Significant)

ak je $p\leq0,05$, je tento test štatisticky preukazný na hladine 95%, dáva sa jeden krížik

ak je $p\leq0,01$, je tento test štatisticky vysoko preukazný na hladine 99%, dávajú sa dva krížiky

ak je $p\leq0,001$, je tento test štatisticky veľmi vysoko preukazný na hladine 99,9%, dávajú sa tri krížiky

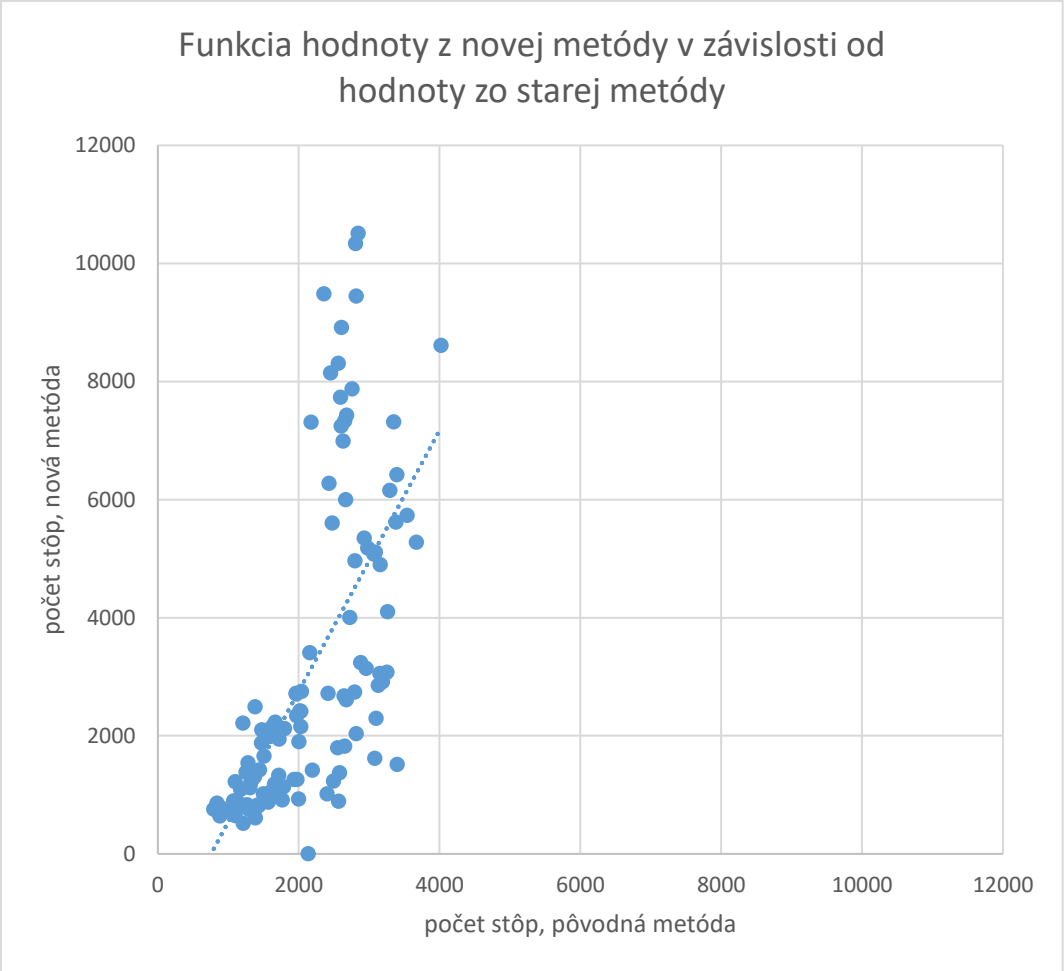
Farebne sú zvýraznené riadky, ktoré označujú jaskyne, kde sa našla zhoda v meraniach preto, že sa nepotvrdil štatistický rozdiel medzi oboma hodnotami. Sú to jaskyne Bystrianska, Domica, Jasovská a Važecká.

Belianska, Demänovská, Driny, Gombasecká, Harmanecká a Ochtinská jaskyňa majú rozdielne merania podľa dvoch metodík. Štatistický rozdiel sa potvrdil a to s vysokou preukaznosťou pri Belianskej, Demänovskej a Ochtinskej jaskyni s interpretom +++. Z toho vyplýva, že časť dozimetrov metódy merajú veľmi podobne a v ďalšej časti sa metódy silno rozchádzajú. To môže byť spôsobené nekvalitou povrchu dozimetra a vysokým počtom falošných detekcií z oboch metód, s tým že každá metóda vyznačuje defekty iným spôsobom a počtom stôp.

Celková p-hodnota zo znamienkového testu je 0,3077 a z Wilcoxonovho testu je 0,51. Pri oboch testoch je výsledok nesignifikantný v zmysle, že rozdiel v meraní súborov dozimetrov z jaskýň nebol potvrdený. Záver z týchto dvoch metód je, že merania sú od seba tak odlišné, že je ťažko vyhodnotiť ich porovnanie, ktorá meria viac stôp, pretože stará alebo nová metóda je zaťažená vysokou chybou, ktorá v oboch prípadoch vzniká rozdielne.

Ďalší spôsob porovnávania ako merajú obe metódy je vytvorenie grafu závislosti hodnôt z novej metódy v závislosti na hodnotách starej metody. (Obrázok 6.6) Na grafe je 115 dátových bodov odpovedajúcich 115 dozimetrom. Dozimeter AH69 bol vyradený ako vysoká odľahlá hodnota. Na osi x sú vyznačené počty stôp na dozimetroch nájdené pôvodnou metódou. Na osi y sú počty stôp z novej metódy. Pri prvom pohľade na graf vidieť, že nová metóda vytvorila približne 13 odľahlých hodnôt v súbore tým, že silno nadhodnotila počty stôp na dozimetroch oproti pôvodnej metóde. Tiež je možné, že v týchto prípadoch spočítala viacej skutočných stôp ako stará metóda.

Krivka je naklonená smerom hore, z dôvodu, že nová metóda ukazuje pri odľahlých dozimetroch viacej stôp ako pôvodná metóda. Bez dozimetrov s vysokými hodnotami sa zdá, akoby niektoré dozimetre predsalen tvorili lineárnu závislosť. Avšak tvar rozloženia bodov na grafe je veľmi rôznorodý, čo sťažuje porovnávanie výstupnej hodnoty metód.



Obrázok 6.6 Hodnota novej metódy v záv. od hodnoty starej metódy.

V hornej časti grafu je vidieť, že časť dozimetrov vybočuje z ideálnej lineárnej závislosti hodnôt nezvyčajne vysokým počtom stôp z novej metódy oproti počtu zo starej metódy. Taktiež vpravo dolu od lineárnej priamky vidieť dozimetre, pri ktorých nová metóda nájde menší počet stôp ako pôvodná metóda.

6.5 Zhodnotenie fungovania detekcie

Pokiaľ dozimetrické sklíčko neobsahuje veľa defektov, potom je zaručená spoľahlivá detekcia stôp novou metódou, väčšina ozajstných stôp je označená s výnimkou stôp na okraji snímky. V tomto ohľade má nová detekcia výhodu oproti starej metóde, ktorá vyznačovala veľké množstvo defektov aj na kvalitnejších snímkach. Avšak niektoré dozimetrické sklíčka mali tak poškriabaný povrch, že nová metóda produkovala veľké množstvo falošných detekcií na niektorých zo 64 segmentov. Z toho vyplýva vysoké zaťaženie automatického merania chybou. To vylučuje možnosť automatického spracovania dozimetrov a aj pri použití novej aplikácie bude treba manuálne korekcie merania. Do akej miery bude obraz spracovávaný manuálne užívateľom, závisí od kvality povrchu daného dozimetra.

Nevyhnutná je vizuálna kontrola vyznačenia stôp algoritmom všetkých 64 snímok, bez ktorej sa nebude dať znížiť veľká kladná odchýlka pri meraní od falošných stôp.

Nová metóda má menej falošných detekcií na snímkach s kvalitnejším povrchom ako pôvodne vyvíjaná metóda a čo je dôležitejšie je, že nová metóda má väčšiu citlivosť, vďaka čomu označuje väčší počet ozajstných stôp, čím sa laborantovi ušetrí čas tým, že ich nebude musieť manuálne vyznačovať v softvéri.

7 GRAFICKÁ APLIKÁCIA

Pre zjednodušenie spracovania snímok z dozimetra bola vytvorená grafická aplikácia, ktorá umožňuje užívateľovi postupne spracovať všetkých 64 snímok z dozimetra. Po vyhodnotení všetkých snímok produkuje výstupný protokol o meraní vo formáte xlsx pre program Microsoft Excel.

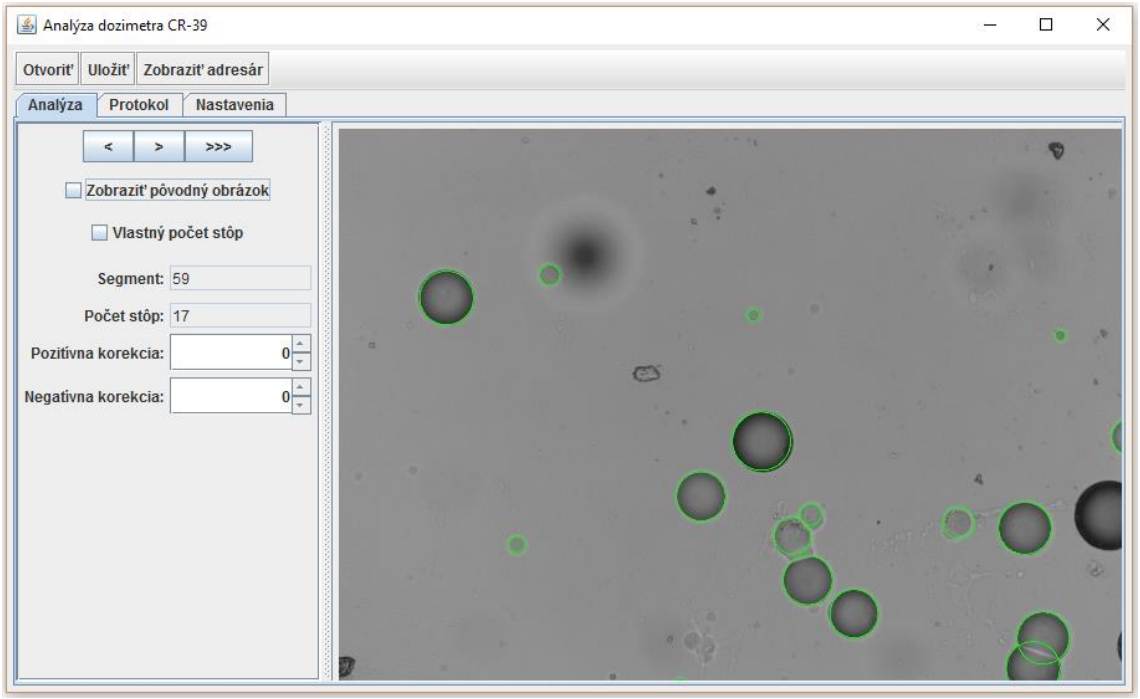
V tejto kapitole bude popísané ovládanie, funkcie a architektúra implementovanej softvérovej aplikácie. Tiež bude popísaný tvar vstupných dát a formáty výstupných súborov.

7.1 Ovládanie

Po spustení aplikácie sa zobrazí jeden hlavný formulár, na ktorom sú dostupné všetky ovládacie prvky. Filozofiou bolo neskrývať podporované funkcie do menu, preto bolo menu nahradené toolboxom nachádzajúcim sa v hornej časti formulára. Takisto je zbytok funkcií rýchlo prístupný cez taby pod toolboxom.

Ako prvý krok užívateľa je otvorenie adresára so snímkami dozimetra pomocou tlačidla *Otvoriť*. Zobrazí sa dialógové okno, ktoré zobrazuje iba existujúce adresáre, pomocou ktorého užívateľ zvolí adresár. Po otvorení adresára sa zobrazí prvý segment dozimetra. Ak si adresár nevyberie, ostávajú otvorené snímky naposledy analyzovaného dozimetra. Na toolbare je tiež tlačidlo *Uložiť*, ktoré exportuje výsledný protokol o meraní vo formátoch xlsx a xml, ktoré funguje až po spracovaní všetkých 64 segmentov dozimetra. Po jeho stlačení sa sa tiež otvorí aplikácia Microsoft Excel s otvoreným novým protokolom, pokiaľ je predvolená pre otváranie súborov xlsx. Tlačidlom *Zobraziť adresár* si užívateľ otvorí okno s adresárom, v ktorom sa nachádzajú vygenerované protokoly.

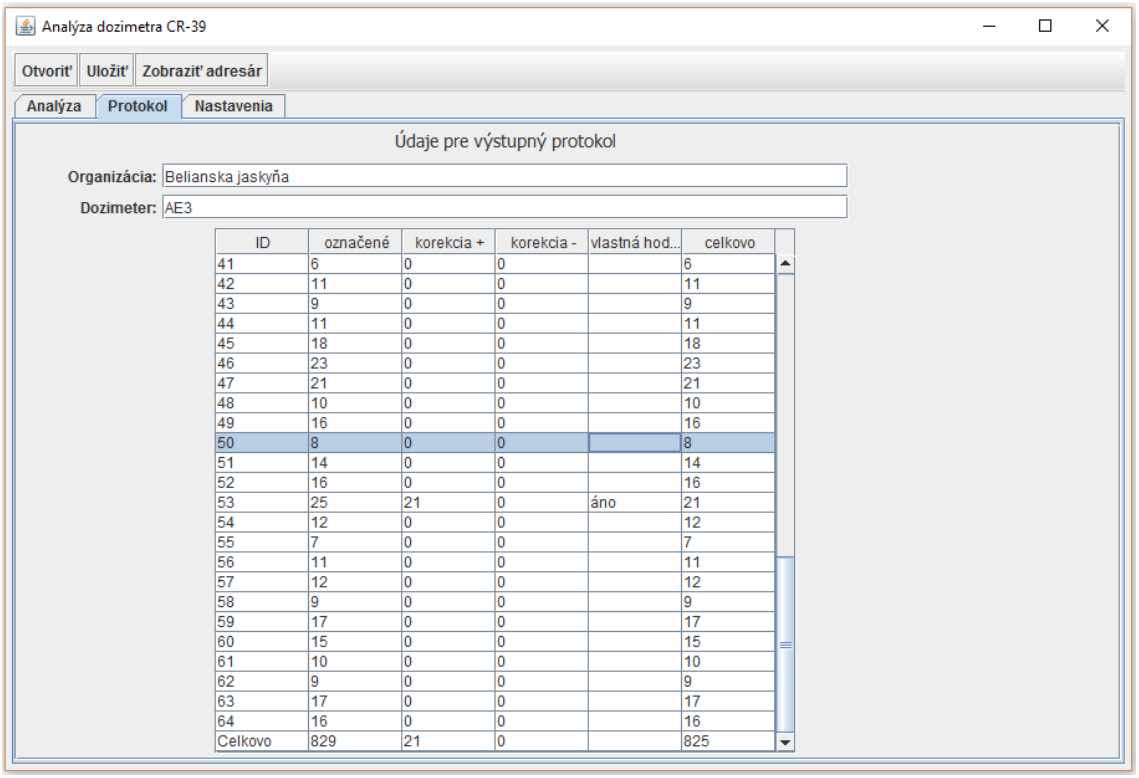
Počas spracovania dozimetra môže užívateľ voľne prepínať medzi tabmi *Analýza*, *Protokol* a *Nastavenia*.



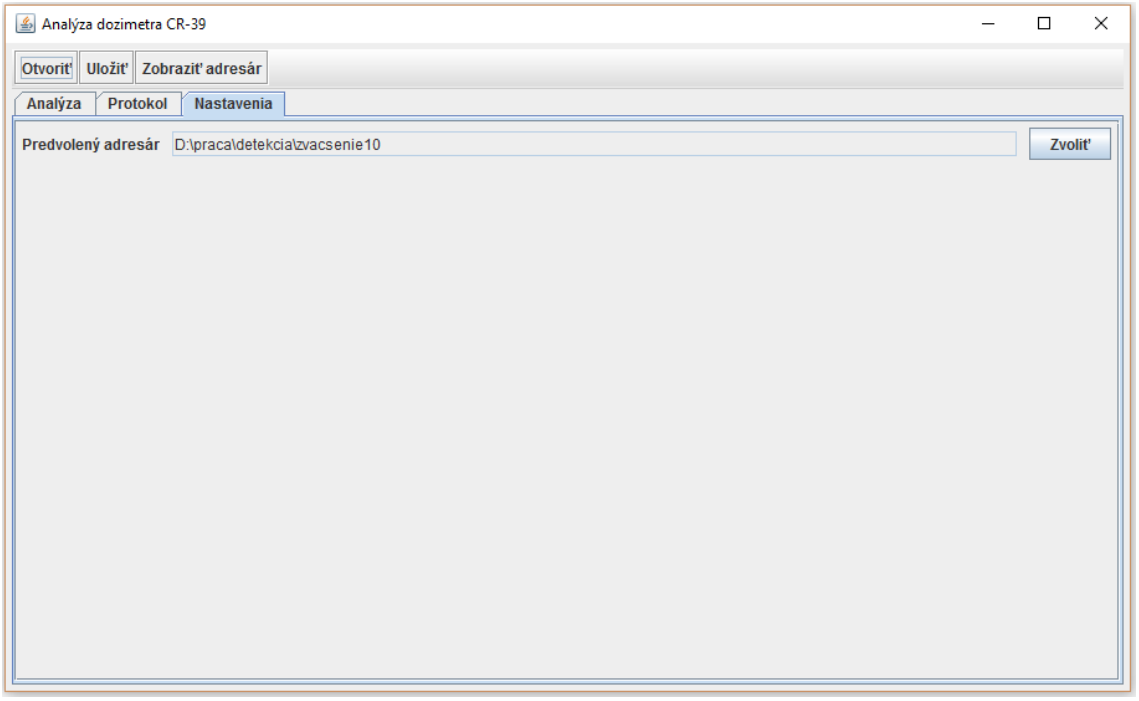
Obrázok 7.1 Hlavný formulár aplikácie otvorený na tabe Analýza.

Snímky sa spracúvajú v tabe *Analýza*. Užívateľ vizuálne skontroluje, či boli správne označené stopy na segmente. Ak chce pridať zopár neoznačených stôp, do kolónky pozitívna korekcia zadá počet stôp, ktoré treba pridať. Do kolónky *Negatívna korekcia* zadá počet falošných stôp. V prípade, že detekcia zlyhá a počet stôp je vysoký, sa použije checkbox „Vlastný počet stôp“, ktorý vyradí detekciu stôp a počet stôp sa zadá manuálne pomocou kolónky *Pozitívna korekcia*. Kolónka *Negatívna korekcia* je vtedy deaktivovaná. Užívateľ môže kedykoľvek stlačiť jedno z troch tlačidiel, ktorými zobrazí ostatné segmenty. Tlačidlom „<“ zobrazí predchádzajúci segment s o 1 nižším ID. Tlačidlo “>” zobrazí následujúci segment s o 1 vyšším ID. Nakoniec tlačidlo „>>>“ urýchľuje analýzu, pretože preskočí na najbližší segment, ktorý ešte nebol detegovaný a spracovaný.

Na tabe *Protokol* sa nachádzajú textové kolónky pre vyplnenie niektorých metadát k meraniu akými sú organizácia, ku ktorej patrí vyvolaný dozimeter a identifikačné sériové číslo dozimetra (kolónka *Dozimeter*.) Nasleduje tabuľka s riadkami odpovedajúcimi segmentom a so stĺpcami, v ktorých sú počty stôp z rôznych krokov metódy. Stĺpec *ID* je identifikačné číslo segmentu od 1 po 64. V stĺpci *označené* je počet stôp označený metódou a užívateľom. *Korekcia +* je stĺpec s hodnotami, ktoré sa majú pripočítať ku konečnému počtu stôp daného segmentu. *Korekcia –* sa odpočítava od počtu stôp segmentu. V stĺpci *vlastná hodnota* sa môže zobraziť “áno”, čo značí že za konečnú hodnotu počtu stôp sa zoberie iba užívateľom zadaná pozitívna korekcia (stĺpec *Korekcia +*). Úplne posledný riadok obsahuje celkové hodnoty pre celý dozimeter. V stĺpci “*celkovo“* v poslednom riadku označenom „*celkovo“* sa nachádza nájdený počet stôp na dozimetri.



Obrázok 7.2 Hlavný formulár aplikácie otvorený na tabe Protokol.



Obrázok 7.3 Hlavný formulár aplikácie otvorený na tabe Nastavenia.

V tabe *Nastavenia* sa zatiaľ nachádza iba nastavenie predvoleného adresára, v ktorom sa budú hľadať adresáre so snímkami. Po stlačení tlačítka *Zvoliť* sa zobrazí

dialógové okno na voľbu predvoleného adresára. Po výbere adresára sa nové nastavenie ihneď uloží na disk. V budúcnosti môžu byť nastavenia rozšírené o ďalšie možnosti.

7.2 Vstupný adresár s obrázkami

V programe sa vyberá adresár so vstupnými obrázkami, ktoré patria k analyzovanému dozimetru. Program v adresári s meraním očakáva 64 obrázkov pre 64 snímkovaných segmentov dozimetra s názvom v tvare IMGx.jpg, kde x je číslo od 1 po 64.

7.3 Formáty výstupných súborov

Program v súčasnej verzii produkuje zatiaľ 2 typy súborov. Prvým typom je excelový súbor, ktorý obsahuje výstupný protokol o meraní s celkovým počtom nájdených stôp a identifikáciou dozimetra.

Druhý typ je xml súbor, ktorý obsahuje detailný výstup z algoritmu, ako sú pozície jednotlivých nájdených kruhov, či boli filtrované, vyradené užívateľom a podobne.Tento súbor má za účel uchovávať údaje o detekcii pre následnú diagnostiku alebo štatistické spracovanie väčšieho množstva vyhodnotených dozimetrov. Pôvodne navrhnutý algoritmus nemal tieto údaje na výstupe, čo znemožnilo lepšie porovnanie kvality detekcie pre celý súbor dozimetrov, ktoré boli spracované v tejto práci. Ďalej v budúcnosti sa bude dať softvérom xml súbor otvoriť a znova prehliadať výsledky detekcie.

7.3.1 Formát výstupného protokolu xlsx

Dáta sú exportované do excelového formátu xlsx. Na pozíciu bunky B2 je vložený titulok. Ďalej zaujímavé údaje sú metadáta k meraniu, konkrétne dátum vygenerovania protokolu v lokálnom čase je v bunke C4, organizácia v ktorej bol použitý dozimeter je v bunke C5, indentifikačné sériové číslo konkrétneho dozimetra je v bunke C6.

Pod touto hlavičkou protokolu je tabuľka, s hlavičkou v riadkoch 10 a 11. Prvý segment sa nachádza v riadku 12. V stĺpci B je identifikačné čislo segmentu od 1 po 64. V stĺpci C je počet stôp nájdených metódou a označených užívateľom. V stĺpci D je kladná korekcia hodnoty a v stĺpci E je záporná korekcia. Stĺpec F obsahuje reťaztec „áno“, pokiaľ bol celkový počet stôp nastavený manuálne užívateľom. Vtedy sa započítava do hodnoty v stĺpci G iba hodnota kladnej korekcie. V stĺpci G je výsledný počet stôp v danom segmente spočítaný z predchádzajúcich stĺpcov.

Riadky pre jednotlivé segmenty sú 12 až 75. Celkový súčet hodnôt za jednotlivé segmenty sa nachádza na riadku 76. V bunke C76 je celkový počet označených stôp, v D76 je celkový súčet kladnej korekcie, v E76 je celkový súčet zápornej korekcie.

Nakoniec v G76 sa nachádza výsledný počet stôp z celého dozimetra. Táto hodnota sa ďalej prepočítava na dávku žiarenia konkrétneho zamestnanca, ktorú absorboval.

Pozície hodnôt sú zaujímavé z toho dôvodu, ak bude potreba automaticky exportovať dáta z väčšej sady protokolov. Dáta sa budú dať získať buď z excelového formátu alebo z formátu xml.

7.3.2 Formát xml

Tento formát bude v budúcnosti použitý na otváranie vytvorených protokolov. Taktiež sa z neho dajú exportovať dáta na vytvorenie štatistiky pre väčší súbor dozimetrov. Prvý riadok je hlavička xml súboru:

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
```

Následuje koreňový tag s názvom protocol. Vnútri tohto tagu sa nachádzajú všetky ostatné tagy. Celková štruktúra je načrtnutá v nasledujúcom výpise:

```
<protocol>
  <trackcount circles="641" final="603" frommethod="603" negativecor="0"
positivecor="0" selected="603"/>
  <segments>
    <segment id="1">
      <detection elapsedtime="591"/>
      <trackcount circles="27" custom="0" final="18" frommethod="18" negativecor="0"
positivecor="0" selected="18"/>
      <circles>
        <c e="1" r="52" x="1125" y="398"/>
        <c e="1" r="57" x="1421" y="816"/>
        <c e="1" r="72" x="816" y="763"/>
      </circles>
    </segment>
    <segment id="2">
        …
    </segment>
  </segments>
</protocol>
```

Vnútri tagu *protocol* sa nachádzajú tagy *trackcount* a *segments*. Tag *trackcount* ukladá sčítané počty nájdených stôp zo všetkých 64 segmentov. Tag *segments* združuje 64 tagov *segment*, ktoré obsahujú identifikáciu segmentu, počty nájdených stôp v danom

segmente a dokonca jednotlivé pozície detegovaných kruhov spolu s informáciou, či bol kruh filtrovaný algoritmom alebo užívateľom.

Tag trackcount. Obsahuje celkom 6 atribútov. Získavajú sa súčtom hodnôt zo všetkých segmentov. Atribút *final* obsahuje konečnú odhadovanú hodnotu počtu stôp na vyšetrovanom dozimetri. Pre diagnostiku sú ukladané ďalšie hodnoty. Atribút *circles* obsahuje počet detegovaných kruhov Houghovou transformáciou. Táto hodnota sa ďalej zníži o kruhy, ktoré boli eliminované algoritmom hľadajúcim prekrývanie sa kruhov a uloží sa ako atribút *frommethod*. V budúcnosti bude môcť užívateľ ručne odznačiť niektoré stopy, čo znižuje hodnotu *frommethod* na hodnotu atribútu *selected*. K tejto hodnote sa ešte pripočíta celková pozitívna korekčná hodnota (*positivecor*) a odčíta negatívna korekčná hodnota (*negativecor*), zadané užívateľom. Výsledkom je hodnota *final*.

Tag segments. Obsahuje 64 tagov segment. Každý z týchto tagov má atribút *id*, ktorý značí identifikáciu segmentu, čo je číslo od 1 po 64.

Tag segment. Obsahuje 3 tagy – *trackcount, circles* a *detection*. Tag *trackcount* obsahuje hodnoty počtu stôp pre daný segment obrazu nájdený na danom obrázku IMGx.jpg, kde x je číslo od 1 po 64 zhodné s *id* segmentu. Takisto ako pri súčtovom tagu *trackcount* vnútri tagu *protocol* aj tu atribút *final* značí výslednú hodnotu počtu stôp. Ostatné atribúty tiež odpovedajú predchádzajúcemu popisu. Pribudol atribút *custom*, ktorého hodnota je 1, ak sa má ako hodnota *final* počítať iba hodnota pozitívnej korekcie (*positivecor*). V opačnom prípade má custom hodnotu 0 a výsledok *final* sa počíta z hodnoty *selected* pripočítaním *positivecor* a odčítaním *negativecor*. Tento prepínač bol vložený aby mal užívateľ možnosť manuálne určiť počet stôp na segmente.

Tag *circles* vnútri tagu *segment* obsahuje zoznam všetkých kruhov, ktoré označila Houghova transformácia. Vnútri tagu *circles* je ľubovoľný počet tagov *c*. Tento tag značí kruh so stredom s 2D súradnicami *x, y*, polomerom *r* a atribút *e* značí, či bol kruh vyradený z výsledku filtráciou prípadne užívateľom.

Údaje obsiahnuté v tagu *segment* sa dajú použiť na opätovné zobrazenie detekcie obrázku a jeho manuálnej korekcie bez potreby opätovného spustenia detekčného algoritmu.

Tag *detection* vnútri tagu *segment* slúži na uchovanie metriky času, atribútu *elapsedtime*, ktorý zabralo načítanie obrázku z pevného disku a následná detekcia pomocou Houghovej transformácie. Hodnota *elapsedtime* je v milisekundách.

7.4 Architektúra

Grafická aplikácia je napísaná v jazyku java a pre svoj beh používa prostredie javy: Java SE 1.8 [29]. Vyvinutá bola pomocou vývojového prostredia eclipse java neon, konkrétne s verziou Neon.2 Release (4.6.2) [30]. Na detekciu kruhov v obraze je použitá už spomínaná knižnica openCV, tu vo verzii 3.2.0. Keďže softvér exportuje protokol

o meraní do formátu XLSX pre program Microsoft Excel, bola použitá tiež knižnica Apache POI vo verzii 3.16 [31], ktorá podporuje otváranie a ukladanie formátu xlsx a xls.

Pri vytváraní aplikácie bola použitá architektúra model view controller, pri ktorej má view čiastočne prístup k modelu, aby mohol užívateľ meniť hodnoty uložené v modeli ako je počet stôp na segmente dozimetra.

Model je najrozsiahlejšia časť aplikácie a obsahuje triedy, ktoré modelujú všetky dáta, čo aplikácia načíta, vytvorí alebo bude ukladať vrátane načítaných obrázkov, počtov nájdených stôp ale aj nastavení aplikácie. Model obsahuje triedy na načítanie nastavení aplikácie, obrázkov, triedy pre vykonanie samotnej detekcie kruhov v obraze, triedu Segment pre modelovanie dát jedného segmetu dozimetra, triedu Measurement ktorá modeluje jeden dozimeter a triedu Protocol, ktorá ukladá metadáta o meraní ako je identifikácia organizácie a dozimetra.

Úlohou view je zobrazovať formulár grafickej aplikácie, na ktorom sa nachádzajú všetky ovládacie prvky, pomocou ktorých sa vizuálne kontroluje výsledok detekcie a umožňuje sa manuálna korekcia detegovaných hodnôt. View obsahuje logiku, pomocou ktorej sa po použití formulárového prvku zmení odpovedajúca hodnota uložená v triede modelu a naopak, logika načítava hodnoty z modelu, ak ich treba zobraziť.

Controller má za úlohu riadiť postupnosť krokov, ktoré prebiehajú pri otváraní adresára so snímkami dozimetra, pri prechádzaní snímok v grafickom rozhraní a pri ukladaní protokolu. Trieda sa volá MainWindowController. Controller za týmto účelom obsahuje stavový automat. Formulár vo view volá niektoré metódy controlleru zakaždým, ak užívateľ chce vykonať nejakú akciu, ako napríklad posun na ďalšie alebo predchádzajúce políčko, posun na ďalšie nespracované políčko a ukladanie protokolu.

Týmito metódami sa controller-u signáluje, že má vykonať danú akciu, k čomu použije model a volania tried v ňom obsiahnuté. Controller takto čiastočne oddeľuje view od modelu, pretože view by sa malo starať o zobrazovanie dát a nemalo vedieť napríklad, ako sa majú dáta uložiť do excelového súboru.

Vstupným bodom do celej aplikácie je trieda *Main*, ktorá obsahuje statickú metódu main, ktorá je volaná pri spustení programu. Tá následne zavolá statickú metódu *createMainWindow()*, v ktorej sa vytvoria inštancie tried AppSettings, MainWindow a MainWindowController. Prebehne pokus o načítanie súboru s nastavením programu. Následne sa triedy sa navzájom poprepájajú a nakoniec sa zobrazí hlavný formulár. Tieto akcie prebehnú vo vlákne event dispatch thread grafickej knižnice java swing. Trieda sa nachádza v package nazvanom *com.dozimeter*.

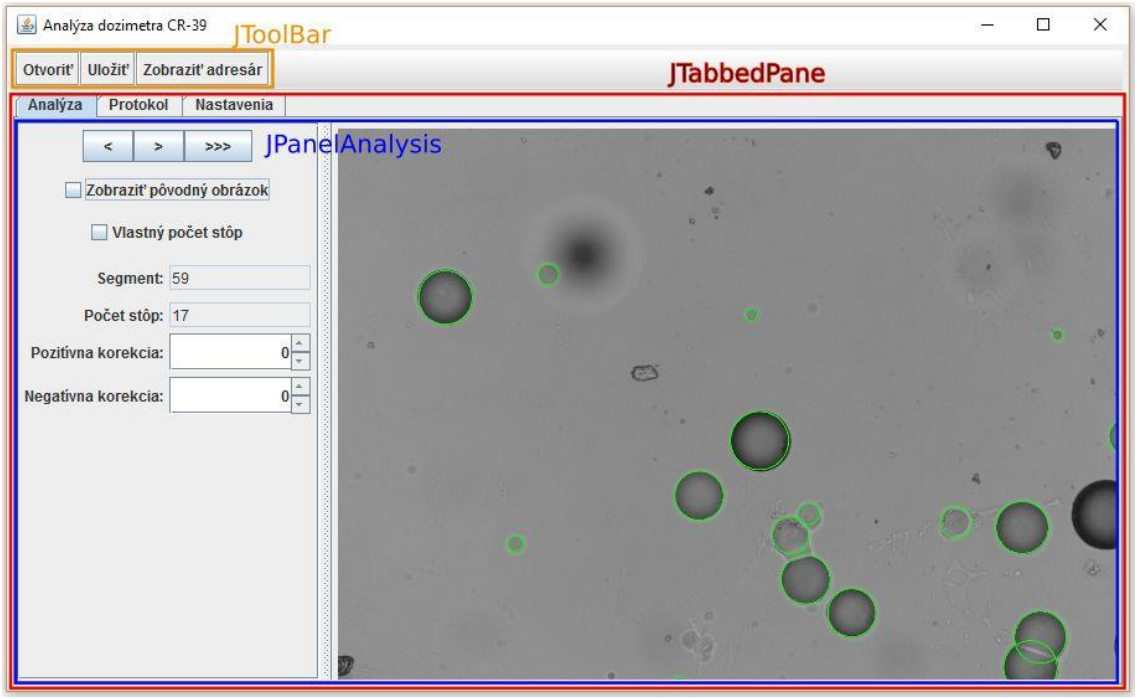
Architektúra je zohľadnená aj v štruktúre zdrojových kódov v Jave. Triedy, ktoré súvisia s grafickým rozhraním sa nachádzajú v package *com.dozimeter.view*. Trieda kontroléru sa nachádza v *com.dozimeter.controller*. Triedy modelu sú v *com.dozimeter.model*. Trieda na exportovanie do Excelu má zvlášť package nazvaný *com.dozimeter.model.export*. Výnimky, ktoré sú vyhadzované rôznymi triedami sa nachádzajú v *com.dozimeter.util.exceptions*.

7.5 View

Grafické rozhranie aplikácie bolo čiastočne designované pomocou pluginu Eclipse WindowBuilder, ktorý automaticky generuje zdrojový kód konštruktora panelu alebo okna aplikácie, kde sa vytvárajú jednotlivé grafické komponenty. Parametre, ktoré sa nedali nastaviť cez grafický designér boli doplnené ručne do kódu.

Aplikácia má jeden formulár, ktorého trieda má typ *MainWindowController* rozširujúci triedu *JFrame*. Po vytvorení inštancie sa volá metóda *void setController(MainWindowController mwc),* ktorou sa priradí kontrolér ku formuláru a tiež metóda *void setAppSettings(AppSettings settings)*, ktorou sa nastaví inštancia nastavení aplikácie, ktorá je využívaná aj kontrolérom.

Panel nástrojov na vrchu aplikácie je typu *JToolBox*. Celú plochu v strede okna vypĺňa *JTabbedPane*, čo je panel so záložkami (tabmi). Jednotlivé taby *Analýza*, *Protokol* a *Nastavenia* majú svôj vlastný design oddelený v zvlášť triedach nazvaných *JPanelAnalysis*, *JPanelProtocol* a *JPanelSettings*. Designujú sa teda zvlášť. Taby sú odvodené od triedy *JPanel*, ktorá sa dá pridať ako tab do *JTabbedPane*.

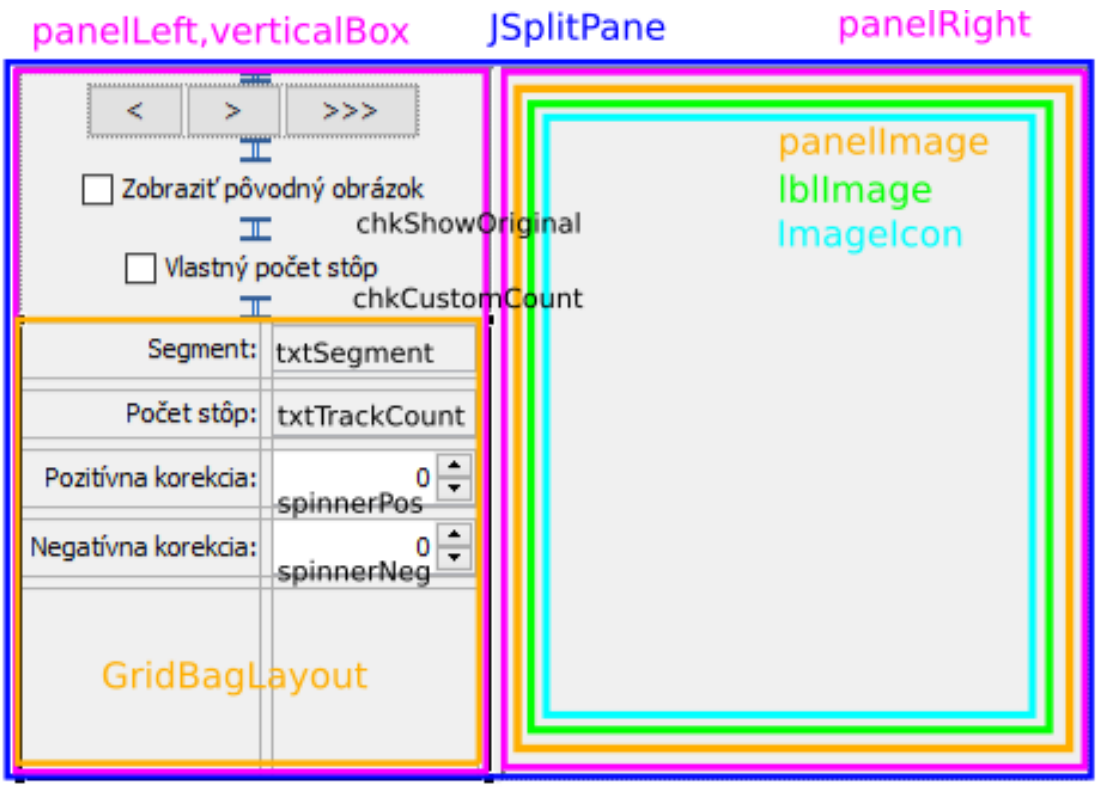


Obrázok 7.4 Rozloženie komponent hlavného formulára aplikácie

Najzložitejší panel je *Analýza* s triedou *JPanelAnalysis*. Celú plochu panelu zaberá komponenta *splitPane* typu *JSplitPane*, ktorá oddeľuje ovládacie prvky vľavo od plochy, v ktorej sa zobrazuje snímka s vyznačenými stopami. Užívateľ môže presúvať preďelujúcu čiaru medzi oboma plochami, vďaka čomu si môže prispôsobovať rozloženie okna. *SplitPane* potrebuje 2 panely typu *JPanel*, ktoré boli nazvané *panelLeft* a *panelRight*. *PanelLeft* obsahuje ovládacie prvky ako tlačítka, ktoré posúvajú snímku

o jednu ďalej alebo dozadu a tiež na najbližšiu nespracovanú. Plochu panelu vypĺňa *JscrollPane*, ktorý umožňuje zobraziť posuvníky, ak sa okno zmenší, až by sa skryli ovládacie prvky. Vnútri je vertikálny Box, ktorý umiestní pod seba komponenty, ktoré bolo do neho vložené. Tlačítka sú v horizontal Box-e. Vertikálne medzery medzi prvkami boli vytvorené pomocou komponenty *verticalStrut*. Pod checkboxami sa nachádza *JPanel* s nastaveným rozložením *GridBagLayout*, vďaka ktorému majú vnútorné komponenty rovnakú šírku a odsadenie. Mriežka má 5 riadkov a 2 stĺpce.

V prvom stĺpci sú popisky ovládacích prvkov v druhom stĺpci. Posledný riadok GridBagLayoutu obsahuje textovú popisku JLabel bez textu, vďaka ktorej sa zabráni roztiahnutiu ostatných komponent. To je zariadené nastavením parametru weighty GridBagConstraint-u, ktorý patrí ku labelu na vysokú hodnotu 20,0.



Obrázok 7.5 Rozloženie komponent panelu JpanelAnalysis

Zobrazenie segmentu na formulári nastane zavolaním metódy void setSegment(Segment seg) triedy MainWindow. Tá zavolá rovnomennú metódu triedy JPanelAnalysis, ktorá sa postará o načítanie obsahov formulárových prvkov z modelu, prostredníctvom parametru seg. Takto si formulár pamätá už predtým zadané hodnoty na predtým zobrazovaných segmentoch.

Tlačítka volajú odpovedajúce akcie popísané v podkapitole 7.6. Ak je zaškrtnutý checkbox *chkShowOriginal*, potom sa nezobrazuje snímka s vyznačenými detegovanými stopami. Ak je zaškrtnuté chkCustomCount, potom sa užívateľovi zneprístupní zmena

negatívnej korekcie a ako počet stôp na snímke sa bude brať iba hodnota z pozitívnej korekcie.

Textové pole txtSegment sa nedá editovať a obsahuje ID aktuálneho segmentu. Textové pole txtTrackCount ukazuje počet stôp, ktorý bol spočítaný uplatnením všetkých korekcií hodnôť ktoré sú prístupné na formulári.

Do spinneru „Pozitívna korekcia“ (premenná *spinnerPos*) sa po zobrazení segmentu na formulári načíta hodnota z inštancie triedy Segment aktuálne zobrazovaného segmentu. Obdobne sa automaticky zaškrtne chkCustomCount, ak bola na segment predtým uplatnený vlastný počet stôp. Po zmene niektorého zo spinnerov sa volá odpovedajúci setter *seg.setPositiveCountCorrection(int positiveCountCorrection)* alebo *seg.setNegativeCountCorrection(int negativeCountCorrection)*, vďaka čomu sa ihneď zmeny údajov ukladajú do modelu. Tiež po zaškrtnutí alebo odškrtnutí chkCustomCount sa volá setter segmentu *void setCustomValueEnabled(boolean customValueEnabled),* ktorý uloží do segmentu, či je manuálne korigovaný.

V pravej časti JpanelAnalysis sa nachádza *panelRight* typu *JPanel*, ktorý slúži na vycentrovanie vloženého panelu *panelImage*. V tomto paneli sa nachádza *JLabel* s názvom *lblImage*, ktorý obaluje inštanciu triedy ImageIcon, vďaka ktorej je zobrazovaná vyznačená alebo originálna snímka. Panely sú uspôsobené tak, aby sa dala v budúcnosti odčítať pozícia pixelu, na ktorý bolo v obrázku kliknuté pre pridávanie editačných funkcií.

Panel *JPanelProtocol* obsahuje tabuľku typu *JTable*. Do tabuľky je priradená inštancia triedy *SegmentsTableModel*, čo je trieda, ktorá má pri vytváraní ako parameter triedu *Measurement*, z ktorej si trieda čerpá dáta o všetkých segmentoch. Takto sa tabuľka napojí na model aplikácie a obsahuje vždy čerstvé informácie, ktoré boli zadané na tabe *Analýza*.

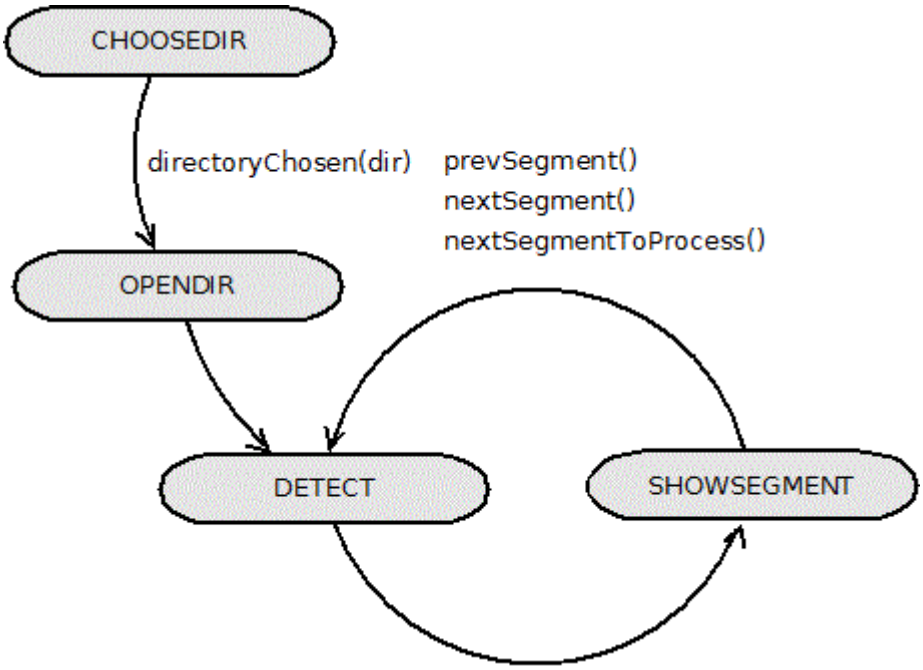
Panelu JPanelSettings sa na začiatku priradí inštancia nastavení aplikácie pomocou metódy *void setAppSettings(AppSettings appSettings)*. V tejto metóde sa obnoví obsah všetkých polí na formulári. Pôvodným zdrojom dát je súbor s nastaveniami, ak bol predtým úspešne načítaný. Ak nebol, použijú sa implicitné nastavenia. Plochu panelu vypĺňa priestor formátovaný pomocou *GridBagLayout*, ktorý sa osvedčil aj na paneli *JPanelAnalysis* na spinnery a textové polia. Pole txtDefaultDirectory zobrazuje, ktorý adresár je predvolený na otváranie adresárov so snímkami. Po stlačení tlačítka *Zvoliť* sa otvorí dialógové okno, ktoré umožňuje zvoliť iba adresár a skrýva súbory. Ak bol zvolený adresár, volá sa odpovedajúci setter v *AppSettings* a následne sa nastavenia hneď ukladajú na disk do súboru s nastaveniami.

7.6 Kontrolér

Túto funkciu plní trieda MainWindowController. Má celkom 2 úlohy. Prvou je oddeľovať grafické rozhranie od funkčnej časti tým, že kontrolér vie ako sa akcie

vykonávajú a rozhranie ich iba spúšťa volaním metód kontroléru. Druhou úlohou je riadenie postupnosti krokov, ktorými sa zobrazujú snímky segmentov na obrazovke pomocou stavového automatu.

Na obrázku je vidieť postupnosť stavov, ktorá bude vysvetlená v nasledujúcom texte:



Obrázok 7.6 Stavový automat pre kontrolér hlavného formulára aplikácie.

Pri volaní konštruktoru sa stavový automat uvedie do stavu CHOOSEDIR. V tomto stave ešte nieje segment na zobrazenie, pretože nebol otvorený žiadny adresár so snímkami. Pred používaním kontroléru sa metódou *void setMainWindow(MainWindow mainWindow)* priradí asociované hlavné okno. Metódou *void setSettings(AppSettings settings)* sa ku kontroléru priradia nastavenia aplikácie.

Až si užívateľ vyberie adresár, formulár zavolá metódu *void directoryChosen(File directory)*. Následne sa stavový automat dostane do stavu *OPENDIR*. V tomto stave sa zavolá metóda *void openDirectory()*. Následne sa vytvoria inštancie tried Measurement a Protocol, ktoré budú popísané v následujúcej kapitole. Overí sa existencia adresára a 64 obrázkov jpeg. Nastaví sa pre spracovanie segment s indexom 0 a automat sa dostane do stavu *DETECT*. Vo zvolenom segmente sa detegujú stopy a potom sa automat dostáva do stavu *SHOWSEGMENT*. V tomto stave sa volá metóda *mainWindow.setSegment*, ktorá spôsobí zobrazenie snímky s vyznačenými detegovanými stopami na formulári v tabe Analýza. V tomto stave ostáva automat do momentu, kým si užívateľ nezvolí iný segment.

Tlačidlu predchádzajúceho segmentu (s popiskom "<") zodpovedá volanie metódy *void prevSegment()*. Zvolí sa segment s indexom o 1 menším a automat sa nastaví

do stavu *DETECT*. V tomto stave nemusí nastať detekcia segmentu, ak už bol detegovaný. Samozrejme sa segment prebehnutou postupnosťou stavov zobrazí na formulári. Tlačidlo následujúceho segmentu (s popiskom „>“) volá metódu *nextSegment()*. Obdobne sa zobrazí ďalší segment v poradí s indexom o 1 viac. Nakoniec metóda *nextSegmentToProcess()* zistí, ktorý segment ešte nebol spracovaný a detegovaný a zvolí ho, čo šetrí užívateľovi čas pri presúvaní sa na ďalšie snímky. Vyvolávaná je tlačidlom pre ďalší nespracovaný segment (s popiskom „>>>“).

Metóda *void saveProtocol()* je volaná tlačidlom *Uložiť* a ak sú všetky segmenty spracované užívateľom, potom prebehnú 3 akcie. Detailné výsledky sa uložia najprv do xml súboru do výstupného adresára *meranie* vnútri adresára so snímkami do súboru s názvom *protokol.xml*. Následne sa dáta exportujú volaním *ExportExcel.ExportProtocol* do excelového workbooku s názvom *protokol.xlsx* v podadresári *meranie*. Nakoniec sa súbor xlsx otvorí predvolenou aplikáciou v systéme, čo má typicky za následok otvorenie programu Microsoft Excel.

Metóda *openResultsDirectory()* je volaná tlačidlom *Zobraziť adresár*. Otvorí sa podadresár *meranie*, ktorý je vnútri adresára so snímkami. Ďalej metóda *nextState* implementuje samotný stavový automat.

7.7 Model

V následujúcej kapitole sú popísané triedy, ktoré sú použité v modeli aplikácie. Triedy spolu ukladajú všetky dáta, ktoré aplikácia načíta zo súborov, zobrazuje, spočíta alebo bude ukladať na disk. Jadrom detekcie obrazu sú triedy *CircleDetection, CircleDetectionResults, CircleDetectionSettings, ImageCV, HoughCircleDetection, HoughCircleDetectionSettings,* ktoré spolu zaistia detekciu kruhov v obraze. Výsledky sa potom spracúvajú a ukladajú v triedach *Protocol*, *Measurement*, *Segment*. Trieda *Protocol* obsahuje jedno meranie *Measurement*, ktoré sa skladá zo 64 tried *Segment*.

AppSettings. Trieda ukladá nastavenia aplikácie. Po jej vytvorení obsahuje implicitné nastavenia ako je implicitná cesta ku nastaveniam a cesta v ktorej sa hľadajú adresáre s nasnímkovanými dozimetrami. Trieda určuje, v ktorom adresári sa budú ukladať nastavenia pomocou metódy *File getSettingsFile()*. Takisto obsahuje metódu *Load(),* pomocou ktorej sa pokúsi otvoriť nastavenia z tejto cesty. Implicitná cesta nastavenia je *C:\dozimeter\settings.xml*. Ak súbor s nastaveniami neexistuje, alebo sa vyskytol problém pri jeho čítaní, vyhodí výnimku. V budúcnosti by mohlo byť ukladanie nastavení vylepšené o nové miesta, kde sa dá uložiť ako je adresár %appdata% vo windowse. Nastavenia sa uložia po zavolaní metódy *Save()*.

Cesta, kde sa nachádzajú snímky dozimetrov sa získava pomocou metódy *File getDefaultDirectory(),* ktorá skúsi použiť adresár v nastaveniach, ak však neexistuje použije sa ako cesta C:\. Setterom *void setDefaultDirectory(File measurementDirectory)* sa nastavuje nový predvolený adresár. Trieda dedí triedu *XmlFile*, vďaka čomu sa

nastavenia ukladajú vo formáte xml. V nastaveniach sa zatiaľ ukladá iba adresár so snímkami, avšak v budúcnosti budú pribúdať rôzne nastavenia. Následuje formát xml súboru:

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<settings>
  <measurementdirectory>D:\praca\detekcia\zvacsenie10</measurementdirectory>
</settings>
```

V tagu *settings* sa nachádza tag *measurementdirectory*, ktorého textovým obsahom je predvolená cesta so snímkami dozimetrov.

CircleDetection. Ide o abstraktnú triedu, ktorá pripravuje návrh aplikácie na prípadnú jednoduchú výmenu detekčnej metódy. Trieda obsahuje abstraktnú metódu *Detect*, ktorá je požadovaná od odvodenej triedy:

public abstract CircleDetectionResults Detect(Mat pic, CircleDetectionSettings settings);

Metóda má ako parameter obrázok uložený v premennej typu *Mat*, čo je vnútorný formát knižnice OpenCV pre ukladanie n-dimenzionálnych matíc, ktorý sa používa aj na ukladanie obrázkov. Taktiež vyžaduje parameter settings, ktorý má ako typ abstraktnú triedu CircleDetectionSettings. Vďaka tomu sa v budúcnosti bude dať vymeniť detekčná metóda s minimálnou zmenou kódu.

CircleDetectionResults. V tejto triede sa ukladajú detegované kruhy. Trieda pri volaní konštruktoru dostane parameter typu Mat, ktorý obsahuje výsledky z Houghovej transformácie, ktoré vracia metóda *Imgproc.HoughCircles* z knižnice OpenCV. Tieto výsledky sa uložia do vnútorných štruktúr triedy. Trieda má metódu *int getCount(),* ktorá vracia počet detegovaných kruhov. Ďalej má trieda gettery a settery, pomocou ktorých sa dá dostať ku všetkým kruhom. Metódy *getCircleX(int index), getCircleY(int index)* vracajú pozíciu nájdeného kruhu na pôvodnom obrázku segmentu. Metóda *int getCircleR(int index)* vracia polomer nájdeného kruhu a metóda *boolean getEnabled(int index)* vracia informáciu, či bol kruh vyradený z výsledkov užívateľom alebo filtráciou. Nakoniec metóda *boolean getFiltered(int index)* vracia informáciu o tom, či bol kruh filtrovaný ako prekrývajúci sa. Obdobne sú v triede settery pre všetky tieto parametre.

CircleDetectionSettings. Táto abstraktná trieda je použitá ako typ triedy, ktorý nastaví parametre pre rôzne typy metód na detekciu kruhov. Zatiaľ sa používa iba jedna detekčná metóda, ale typ je použitý pre prípadné rozšírenie programu do budúcnosti.

HoughCircleDetection. Táto trieda obsahuje 3 metódy pomocou ktorých sa detegujú stopy v obraze. Najdôležitejšia je metóda detegujúca kruhy v obraze nazvaná Detect:

public CircleDetectionResults Detect(Mat pic, CircleDetectionSettings settings)

Metóda má ako vstup obrázok typu *Mat* z knižnice OpenCV a nastavenia pre detekciu typu *CircleDetectionSettings*, ktoré sa však interne pretypujú na triedu *HoughCircleDetectionSettings*, v ktorej sú parametre špecifické pre Houghovu transformáciu. Obrázok sa najprv rozmaže mediánovým blurom s maticou veľkosti 9x9 pomocou metódy *Imgproc.medianBlur* z knižnice OpenCV. Následne sa z farebného formátu BGR konvertuje obrázok na odtiene sivej pomocou metódy *Imgproc.cvtColor*. Následne sa volá metóda *Imgproc.HoughCircles*, ktorá nájde kruhy v obraze. Parametre dostane z triedy *HoughCircleDetectionSettings*. Táto metóda vracia zoznam nájdených kruhov v type Mat. Avšak k dátam sa v ňom pristupuje cez indexy, takže je výsledok pre jednoduchšiu prácu s hodnotami uložený do novej inštancie triedy *CircleDetectionResults*. Tá už má gettery k jednotlivým nájdeným hodnotám. Tieto výsledky ešte obsahujú prekrývajúce sa kruhy a preto sú ešte spracované pomocou metódy *Filter* a nakoniec vrátené.

Metóda *HoughCircleDetection.Filter* implementuje algoritmus hľadajúci prekrývajúce sa kruhy:

public static CircleDetectionResults Filter(CircleDetectionResults cdr)

Metóda má ako vstup inštanciu triedy *CircleDetectionResults*, ktorú nakoniec vracia naspäť s novými nastaveniami pre jednotlivé kruhy v zozname. Kruhy, ktoré sú eliminované majú nastavený príznak *filtered* na true a príznak *enabled* je nastavený na false.

Ďalšia je metóda *Mark*, ktorá sa používa na vyznačenie nájdených kruhov v pôvodnom obraze:

public static Mat Mark(Mat pic, CircleDetectionResults cdr, boolean showFiltered)

Metóda má ako vstup zoznam kruhov cdr a tiež prepínač *showFiltered*, ktorý určí, či budú zobrazené aj kruhy, ktoré sa prekrývali s inými, vyradené metódou Filter. Kruhy, ktoré neboli filtrované a majú nastavený príznak enabled sa zobrazia ako kruh zelenej farby. Kruhy, ktoré neboli filtrované ale boli odznačené užívateľom, sa označia hnedou farbou. Nakoniec sa filtrované kruhy vykreslia červenou farbou, čo sa použilo pri testovaní dátovej sady z jaskýň. Nový obrázok je vracaný znova s typom *Mat*.

HoughCircleDetectionSettings. Trieda obsahuje špecifické nastavenia pre houghovu transformáciu. Obsahuje 2 konštruktory, jeden bez parametrov, ktorý vytvorí triedu s prednastavenými parametrami. Ďalší konštruktor má ako parametre všetky nastavenia.

V tabuľke sú vidieť nastavenia, ktoré sú použité vo finálnom programe a takisto boli použité pre vyhodnotenie dátovej sady pri porovnávaní metód:

Prednastavené konštanty	Hodnota	Getter
int DEFAULT_BLURMATRIXSIZE	9	getBlurMatrixSize()
double DEFAULT_DP	1	double getDp()
double DEFAULT_MINDIST	20	double getMinDist()
double DEFAULT_PARAM1	40	double getParam1()
double DEFAULT_PARAM2	30	double getParam2()
int DEFAULT_MINRADIUS	10	int getMinRadius()
int DEFAULT_MAXRADIUS	85	int getMaxRadius()

Trieda tiež obsahuje gettery a settery pre všetky parametre a tiež trieda dedí od triedy XmlFile, preto sa bude dať v budúcnosti nastavenie načítať zo súboru alebo ho ukladať.

ImageCV. Trieda abstrahuje načítanie obrázku pre detekciu obrazu. Volaním konštruktora tejto triedy *ImageCV(File file)* sa v triede uloží názov súboru obrázku s plnou cestou. Po zavolaní metódy *Load()* sa obrázok načíta pomocou metódy Imgcodecs.imread z knižnice OpenCV. Uložené dáta obrázku sú typu *Mat*. K dátam sa dá dostať pomocou metódy *Mat getOriginal()* a takisto sa dá zistiť, či bol obrázok už načítaný pomocou metódy *boolean isLoaded()*.

ISegment. Tento interface zaisťuje, aby sa dali v budúcnosti čítať xml súbory uložené na disku a so staršími verziami dátového formátu. Pri spracovaní dátovej sady jaskýň tiež vznikol neaktuálny formát xml, v ktorom boli uložené výsledky detekcie kruhov a počty stôp. Pre zhodný prístup k týmto dátam bolo vytvorený tento interface.

Segment. Trieda abstrahuje jeden segment na dozimetri a teda jednu snímku dozimetra. Trieda obsahuje obrázok *ImageCV*. Ukladá si nastavenia v triede typu*CircleDetectionSettings* a po vyhodnotení ukladá výsledky detekcie v triede typu *CircleDetectionResults*. Trieda ukladá identifikáciu segmentu (*id*), čo je číslo od 1 po 64. Pamätá si tiež, či bol segment už detegovaný, a či bol spracovaný užívateľom v grafickom rozhraní. (je nutné ho zobraziť). Trieda si tiež ukladá ďalšie informácie o segmente ako je užívateľom nastavená pozitívna alebo negatívna korekcia počtu stôp, ktoré sa pripočítavajú a odčítavajú od vyznačeného počtu stôp z metódy. Ukladá si aj čas, ktorý trvalo načítanie obrázku z disku a detekcia.

Nasleduje popis použitia triedy. Konštruktor triedy má 2 parametre:

public Segment(File pictureFile, int segid)

Parameter *pictureFile* je názov súboru s obrázkom aj s plnou cestou, ďalej *segid* je číslo segmentu od 1 po 64. Segment si tieto údaje uloží a označí sa za nedetegovaný a za nespracovaný užívateľom. Kontroluje sa, či existuje súbor s obrázkom. Ak neexistuje, potom je vyvolaná výnimka *FileNotFoundException*. Následne sa zavolá metóda *Detect()*. Najprv sa načíta obrázok do inštancie triedy *ImageCV*. Použije sa inštancia triedy *HoughCircleDetection* na detekciu stôp v obraze. Automaticky sú filtrované prekrývajúce sa kruhy. Výsledky detekcie sa ukladajú do inštancie triedy *Segment*. Zároveň sa segment označí ako detegovaný a takisto je meraný čas behu metódy *Detect* a uložený do atribútu *elapsedtime*. Po detekcii je možné zavolať metódu *Mat Mark()*, ktorá vyznačí nájdené stopy v obrázku. Obrázok sa môže zobraziť v grafickom rozhraní. Po detekcii je tiež možné volať rôzne gettery, ktoré na základe uloženého zoznamu stôp spočítajú celkové počty kruhov a stôp po rôznych krokoch filtrácie a korekcie užívateľom.

Trieda *Segment* tiež implementuje metódu *Node toXmlNode(Document doc)* dedenú od *XmlFile*, vďaka ktorej sa dajú výsledky detekcie uložiť do súboru xml. Metóda vráti nódu, ktorá sa dá pripojiť do väčšieho stromu tagov xml.

Následuje zoznam getterov triedy Segment na spočítaný počet stôp na segmente:

Getter	Popis vrátenej hodnoty
int getCircleCount()	Počet kruhov, ktoré našla Houghova transformácia v danom segmente. Obsahuje prekrývajúce sa kruhy.
int getMethodTrackCount()	Počet stôp odhadnutých metódou v segmente. Tiež počet kruhov zmenšený o filtrované kruhy.
int getSelectedTrackCount()	Počet stôp, ktoré neboli odznačené užívateľom. Tiež počet z getMethodTrackCount() zmenšený o odznačené stopy.
int getPositiveCountCorrection()	Pozitívna korekcia počtu stôp od užívateľa pre daný segment.
int getNegativeCountCorrection()	Negatívna korekcia počtu stôp od užívateľa pre daný segment.
int getFinalTrackCount()	Konečný počet stôp pre segment korigovaný užívateľom. Ak je zadaná užívateľom vlastná hodnota, bude vrátená. Ak nie hodnota sa získa spočítaním označených stôp a korekcií.

Measurement. Trieda abstrahuje obrazovú analýzu jedného dozimetra. Trieda ukladá 64 inštancií typu Segment, ktoré ukladajú údaje o všetkých segmentoch na dozimetri.

Použitie triedy začína volaním konštruktora spolu s parametrom cesty k adresáru, v ktorom sú analyzované obrázky segmentov:

public Measurement(File directory)

Trieda si uloží cestu k adresáru. Následne treba zavolať metódu *loadDirectory()*. Skontroluje sa, či existuje adresár s obrázkami. Vytvoria sa adresáre pre výstupné súbory. Priamo v adresári s pôvodnými obrázkami je vytvorený adresár *meranie* a v ňom je vytvorený podadresár *vystup*. Do podadresára sa nahrávali obrázky pri spracovávaní dátovej sady z jaskýň. Pri vytváraní inštancií segmentov sa môže stať, že niektorý obrázok neexistuje. Pri niektorej z uvedených chýb sa ďalej vyvolá výnimka *CannotOpenMeasurementException*. Na začiatku je zvolený segment s indexom 0, ktorý má ID 1. K tomuto segmentu sa dá dostať cez volanie metódy *Segment getSelectedSegment()*. Tiež sa dá zistiť index zvoleného segmentu pomocou *int getSelectedSegmentIndex()*. Vybrať iný segment sa dá pomocou volania *void selectSegment(int index)*. Nápoveda, ktorý segment ešte nebol detegovaný a spracovaný užívateľom sa získa volaním *int getNextIndex()*. Metóda vracia najnižší nespracovaný index. To, či sú už všetky segmenty spracované, sa dá zistiť volaním *boolean isEverythingProcessed()*.

Až sú všetky segmenty spracované, trieda *Measurement* obsahuje dáta všetkých segmentov, ktoré sa dajú uložiť do xml alebo exportovať do Excelu pomocou inštancie triedy *Protocol*. Deje sa to pomocou getterov, ktoré spočítajú súčet rôznych druhov počtov stôp zo všetkých 64 segmentov.

Následuje tabuľka getterov triedy Measurement pre počty stôp na celom dozimetri:

Getter	Popis vrátenej hodnoty
int getCircleCountTotal()	Počet kruhov, ktoré našla Houghova transformácia vo všetkých segmentoch, obsahuje prekrývajúce sa kruhy.
int getMethodTrackCountTotal()	Počet stôp odhadnutých metódou vo všetkých segmentoch. Tiež počet kruhov zmenšený o filtrované kruhy.
int getSelectedTrackCountTotal()	Počet stôp, ktoré neboli odznačené užívateľom. Tiež počet z getMethodTrackCountTotal() zmenšený o odznačené stopy.
int getPositiveCorTotal()	Pozitívna korekcia počtu stôp od užívateľa sčítaná zo všetkých segmentov.
int getNegativeCorTotal()	Negatívna korekcia počtu stôp od užívateľa sčítaná zo všetkých segmentov.
int getFinalTrackCountTotal()	Konečný počet stôp pre celý dozimeter korigovaný užívateľom. Hodnota sa získa sčítaním konečného počtu stôp nájdených na jednotlivých segmentoch.

Pri spracovaní dátovej sady jaskýň sa volala metóda *void processAll()*, ktorá spracovala všetkých 64 segmentov naraz a ukladala obrázky s vyznačenými stopami do podadresára *meranie\vystup* v adresári so snímkami.

Protocol. Trieda ukladá metadáta o meraní ako je identifikátor dozimetra a organizácia, v ktorej bol dozimeter použitý. V budúcnosti je možné, že sa rozšíri množstvo metadát, ktoré sa ukladajú vo výstupnom protokole. Triede sa tiež priradí meranie typu *Measurement*. Trieda zaisťuje ukladanie do výstupného formátu xml, pretože dedí od *XmlFile* a implementuje *toXmlNode*. Okrem toho určuje, ako sa budú volať výstupné súbory pre jeden protokol okrem prípony, ktorá bude pri každom formáte iná.

Následuje zoznam getterov a setterov:

Parameter	Getter	Setter
Názov súboru bez prípony	String getFilename()	void setFilename(String filename)
Organizácia pracovníka	String getOrganisation()	void setOrganisation(String organisation)

Sériové číslo dozimetra	String getDosimeter()	void setDosimeter(String dosimeter)

XmlFile. Ide o triedu, od ktorej dedí metódy viacej ostatných tried, ktoré potrebujú ukladať svoje dáta do XML formátu. Obsahuje dôležité metódy Load(File file) a Save(File file). Trieda ich implementuje a netreba ich preťažovať v dcérskych triedach. Tieto metódy vedia načítať alebo uložiť dáta do súboru vo formáte xml. Na prácu s XML je použitý xml DOM (document object model) vstavaný v jave. [32] Používa sa na parsovanie existujúcich xml súborov do triedy *Document*, obsahujúcej strom xml nód, z ktorého sa postupne nachádzajú nódy podľa ich názvu a polohy v štruktúre a čítajú sa dáta v nich obsiahnuté. Takisto sa pomocou DOM vytvára stromová štruktúra xml súboru, ktorý bude zapísaný.

Každá jedna dcérska trieda, ktorá dedí po *XmlFile* musí implementovať abstraktné metódy *toXmlNode* a *fromXmlNode.* Metóda *Node toXmlNode(Document doc),* vytvára stromovú štruktúru xml, ktorá sa bude ukladať do xml súboru. Metóda *void fromXmlNode(Element elem)* číta dáta z naparsovaného xml do svojich vnútorných atribútov triedy. Triedy môžu navyše obsahovať ako atribút inú triedu odvodenú od *XmlFile*, vďaka čomu sa dá modelovať stromová štruktúra xml pomocou stromu tried, ktoré modelujú ukladané dáta. Napríklad trieda *Protocol* obsahuje ako atribút triedu *Measurement,* a tá má ako atribút pole so 64 triedami typu *Segment*. Po použití metódy *Save* na *Protocol* si trieda v metóde *toXmlNode* zavolá *toXmlNode* triedy *Measurement* a výslednú nódu s tagom „segments“ a všetkými ďalšími vnorenými tagmi si pripojí do vznikajúceho xml ukladajúceho celý protokol. Nóda *segments*, ktorú vrátila metóda *measurement.toXmlNode* vznikla vytvorením tagu „segments“, do ktorého bolo postupne vložených 64 nód s tagom „segment“, ktoré boli vytvorené volaním *toXmlNode* jednotlivých inštancií triedy *Segment*.

Trieda obsahuje tiež metódy, ktoré abstrahujú ukladanie a čítanie javových typov ako je int, long, boolean, String, double z xml atribútov pomocou statických metód. Pre ukladanie atribútov slúžia metódy *addIntAttribute, addLongAttribute, addDoubleAttribute, addStringAttribute, addBooleanAttribute*. Ako príklad je metóda:

public static Element addIntAttribute(Document doc, Element elem, String attrName, int value)

Táto metóda berie ako vstupný parameter vytvorený dokument doc, ktorý sa používa pre vytváranie všetkých tagov, ktoré budú patriť do jedného xml súboru. Ďalší parameter je už vytvorený element elem, ktorému bude pridaný parameter s názvom attrName a do ktorého bude uložená hodnota value. Ostatné metódy majú prvé 3 parametre rovnaké a 4-tý parameter má iba iný typ, vždy odpovedajúci názvu metódy.

Atribúty sa čítajú pomocou statických metód *getIntAttribute, getLongAttribute, getDoubleAttribute, getStringAttribute, getBooleanAttribute*. Metódy majú iba 2 parametre. Prvý parameter je element, alebo tiež xml tag z ktorého bude prečítaný atribút s názvom attrName. Návratová hodnota je v súlade s názvom metódy, napríklad getStringAttribute vracia typ String. Nasleduje príklad takejto metódy:

public static int getIntAttribute(Element elem, String attrName)

ExportExcel. Táto trieda zaisťuje export dát do excelového súboru. Obsahuje statickú metódu:

public static File ExportProtocol(Protocol protocol, File directory)

Pomocou tejto metódy sa uložia exportované údaje z triedy protocol do súboru, ktorý je daný kombináciou cesty k adresáru *directory*, názvu súboru, ktorý vracia trieda Protocol a prípony ".xlsx", ktorá je daná konštantou v triede ExportProtocol. Takýmto spôsobom sa dajú napísať ďalšie exportovacie triedy, ktoré budú ukladať súbory do jedného adresára vždy s potrebnou príponou súboru, podľa výstupného formátu súboru. Trieda používa spomínanú knižnicu Apache POI 3.16 a jej funkcionalitu POI-XSSF na exportovanie xlsx súboru, čo je excelový workbook.

7.8 Plánované vylepšenia do budúcnosti

Detekcia celkovo funguje pomerne spoľahlivo na veľkom počte segmentov z rôznych dozimetrických sklíčok, avšak sa nájdu niektoré segmenty, ktoré vykazujú vysoký počet falošných detekcií a je väčší počet segmentov, ktoré obsahujú defekt iba v jednej časti snímky. To by sa mohlo v budúcnosti riešiť skúšaním ďalších metód predspracovania obrazu, ktoré sa robí pred spustením Houghovej transformácie. Jeden z takýchto nápadov je pred samotným rozmazaním obrázku urobiť morfologickú eróziu nasledovanú morfologickou dilatáciou, ktoré možno odstránia malé defekty v obraze oveľa účinnejšie ako obyčajné rozmazanie obrazu. Problém s týmto prístupom je, že sa môžu stratiť stopy, ktoré nie sú také výrazné a kontrastné na snímke.

Ďalšou metódou predspracovania obrazu môže byť vytvorenie masky obrazu, ktorá pravdepodobne obsahuje väčšie objekty, a to skopírovaním pôvodného obrázku nasledovaným morfologickou operáciou erózie a dilatácie, ktorá odstráni jemné detaily a následným thresholdingom výslednej hodnoty pixelov, ktoré musia mať menší alebo väčší jas ako najbližšie pozadie o stanovenú hodnotu. Takto sa získa maska objektov, ktorých tiene vystupujú z nerozoznateľného objemu polymérového materiálu, ktorý sa javí ako jednoliata sivá plocha. Maska by sa mohla dilatovať, aby zhutneli „machule",

ktoré značia vyberané objekty, aby bola väčšia šanca, že pokrýva kompletné objekty. Pomocou masky sa z pôvodného obrázku vyrežú v spracovanom obrázku iba výrazné a väčšie objekty a zvyšok obrázku sa nahradí silne rozmazaným pozadím pôvodného obrázku. Je nutné vyriešiť problém s ostrým prechodom medzi objektami vybranými maskou a medzi nahradeným zmazaným pozadím.

Aplikácia umožňuje manuálnu korekciu počtu stôp zadávaním čísiel do textových kolóniek, avšak by bolo efektívnejšie aby mal užívateľ grafické nástroje, ktoré pomáhajú počítať stopy. Existuje niekoľko nápadov, aké nástroje by sa mohli používať. Napríklad kliknutím myšou na obrázok by sa pridala nedetegovaná stopa. Počet stôp by sa zvýšil o jednu. Nástrojom guma by sa odznačili stopy, ktoré sa nachádzajú v ploche tohto nástroja po stlačení tlačítka myši. Nástroj polygón by umožňoval vybrať oblasť tvaru štvoruholníku, v ktorej by sa odznačili stopy. To by fungovalo ako efektívné riešenie na segmenty, ktoré obsahujú veľké množstvo falošných stôp na jednom ohraničenom mieste.

Pre implementovanie týchto nástrojov by bolo potrebné doplniť funkcionalitu do view, ako sú tlačidlá umožnujúce voľbu použitého nástroja. Ďalej by bolo treba doplniť model aplikácie, aby sa dali ukladať zmeny vykonané nástrojmi. Principiálne by stačilo ukladať miesta označených nových stôp.

Výpočet stanovenej dávky príkonu žiarenia zamestnanca by mohol byť spočítaný pomocou softvéru a mohol by byť súčasťou výstupného protokolu.

Používaním aplikácie v praxi sa zistia požiadavky užívateľov a podľa toho budú upravované a pridávané funkcie softvéru.

ZÁVER

V tejto práci bolo testované použitie Houghovej transformácie obrazu na počítanie kruhových stôp alfa častíc na povrchu polymérového dozimetru typu CR-39. Za týmto účelom bola získaná testovacia sada snímok povrchu dozimetrov, ktorými sa sledovala absorbovaná dávka žiarenia jaskyniarmi v 10 slovenských jaskyniach za 3. štvrťrok roku 2015. Súčasťou sady sú výsledky predtým použitej detekcie obrazu metódou Watershed.

Implementácia Houghovej transformácie pochádza z knižnice OpenCV. Najprv boli ladené parametre Houghovej transformácie tak, aby metóda vyznačovala čo najviac skutočných stôp, a aby bola necitlivá na defekty v obraze, ako sú škrabance na povrchu dozimetra. Boli identifikované optimálne parametre uvedené v kap. 5. Ladenie bolo úspešné, avšak metóda je stále citlivá na viac poškriabané dozimetre, avšak množstvo falošných detekcií je celkovo menšie ako pri pôvodnej metóde. Nová metóda taktiež vyznačuje viac skutočných stôp, čím jej stúpla citlivosť.

Následne bola metóda spustená na kompletnú sadu 116 dozimetrov, ktoré majú spolu 7424 snímok segmentov na dozimetroch (64 snímok na dozimeter). Z vyhodnotenia obrazového výstupu bolo zistené, že aj jeden dozimeter sa líši kvalitou povrchu jeho jednotlivých segmentov. Napríklad 2 zo 64 segmentov vyprodukujú veľký chybný počet stôp. To znamená, že sa nebude dať použiť automatické spracovanie dozimetrov novou metódou bez manuálneho zásahu užívateľom a jeho manuálnej korekcie výsledku. Dozimeter bude možné spracovať s manuálnou korekciou počtu stôp niekoľkých segmentov.

Pre možnú opravu metódy a zníženie citlivosti na defekty v obraze boli navrhnuté možné riešenia ako je použitie morfologických operácií obrazu ako je erózia a dilatácia na odstránenie jemných detailov zo snímky ešte pred použitím Houghovej transformácie na spočítanie počtu stôp. Problém, ktorý komplikoval

Keďže bol v dátovej sade dostupný aj počet stôp identifikovaný pôvodnou metódou, bol porovnaný s výsledkami z novej metódy párovým testom. Výsledky z oboch metód vykazovali rôznorodé odchýlky z dôvodu, že boli v oboch prípadoch zaťažené chybou spôsobenou defektami v obraze, ktoré boli započítané ako stopy, čo viedlo k vyššiemu počtu stôp ako skutočnému. Výsledky dozimetrov z niektorých jaskýň mali blízku hodnotu, avšak výsledky z iných jaskýň sa u oboch metód značne líšili, čo znemožňuje zmysluplné porovnanie výsledkov parametrickými štatistickými metódami. Z frekvenčného polygónu rozloženia hodnôt z dozimetrov sa dá zistiť, že nová metóda počíta pri veľkom množstve dozimetrov menej stôp ako pôvodná, čo sa dá vysvetliť tým, že nová metóda označuje menej falošných stôp na väčšine dozimetrov s výnimkou odľahlých prípadov.

Nová metóda má menej falošných detekcií ako pôvodne vyvíjaná metóda a čo je dôležitejšie je, že nová metóda má väčšiu citlivosť, vďaka čomu označuje väčší počet

ozajstných stôp, čím sa laborantovi ušetrí čas tým, že ich nebude musieť manuálne vyznačovať v softvéri.

Výsledky novej metódy boli prezentované v laboratóriu, a na snímky s vyznačenými stopami bol pozitívny ohlas v tom zmysle, že v praxi bola pôvodná metóda menej citlivá na stopy a označovala väčšie množstvo defektov.

Nakoniec bola vyvinutá aplikácia s grafickým rozhraním, pomocou ktorej sa dá postupne spracovať 64 snímok dozimetra a korigovať počty nájdených stôp. Užívateľ zobrazuje jednu snímku za druhou a vizuálne kontroluje výsledok detekcie. Aplikácia spočíta celkový súčet stôp zo všetkých snímok a produkuje výstupný protokol vo formáte xlsx pre Microsoft Excel. Obsahuje detailný rozpis počtu stôp na jednotlivých segmentoch spolu s hodnotami manuálnych korekcií. Aplikácia bola napísaná v jazyku Java pomocou vývojového prostredia Eclipse a je určená pre 64 bitovú platformu Windows pre verzie 7 až 10.

Literatúra

[1] CABÁNEKOVÁ, H. a D. NIKODEMOVÁ. Usmerňovanie ožiarenia obyvateľstva radónom v pobytových priestoroch [online]. Bratislava: Environment, 2013 [cit. 2016-9-22]. ISBN ISBN 978-80-89384-05-1. Dostupné z: http://www.szu.sk/userfiles/file/Katedry/kat_153/BROZURA.pdf

[2] CR-39. In: Wikipedia: the free encyclopedia [online]. San Francisco (CA): Wikimedia Foundation, 2012 [cit. 2016-9-22]. Dostupné z: https://en.wikipedia.org/wiki/CR-39

[3] Solid state nuclear track detector. In: Wikipedia: the free encyclopedia [online]. San Francisco (CA): Wikimedia Foundation, 2016 [cit. 2016-9-22]. Dostupné z: https://en.wikipedia.org/wiki/Solid-state_nuclear_track_detector

[4] BHAGWAT, A.M. *Solid State Nuclear Track Detection: Theory and Applications* [online]. Kalpakkam: Indian Society for Radiation Physics, 1993 [cit. 2017-05-23]. Dostupné z: http://www.iaea.org/inis/collection/NCLCollectionStore/_Public/25/019/25019093.pdf

[5] Nuclear track detectors. The MoEDAL experiment [online]. Geneva: Cern, 2016 [cit. 2016-09-22]. Dostupné z: http://moedal.web.cern.ch/content/nuclear-track-detectors

[6] KALSI, P.C., A. RAMASWAMI a V. K. MANCHANDA. Solid state nuclear track detectors and their applications. In: *BARC Newsletter* [online]. Vol. 257. Trombay: Bharba Atomic Research Centre, 2005 [cit. 2016-12-10]. ISSN ISSN - 0976-2108. Dostupné z: http://barc.ernet.in/publications/nl/2005/200506-2.pdf

[7] CR-39 technology. LANDAUER [online]. Oxford: LANDAUER, 2015 [cit. 2016-12-10]. Dostupné z: http://www.landauer.co.uk/neutron_cr-39.html

[8] CHUDÝ, M. Základy dozimetrie žiarenia [online]. 2008. Bratislava: Univerzita Komenského v Bratislave [cit. 2016-9-26]. Dostupné z: http://www.dnp.fmph.uniba.sk/~kollar/martin/martin_all.pdf

[9] Indoor radon measurements by nuclear track detectors: applications in secondary schools. FACTA UNIVERSITATIS: Series: Physics, Chemistry and Technology [online]. Beograd: University of Niš, 2006, 2006(4), 93-100 [cit. 2016-12-10]. ISSN 2406-0879. Dostupné z: http://www.doiserbia.nb.rs/img/doi/0354-4656/2006/0354-46560601093B.pdf

[10] Oddelenie radiačnej hygieny. Slovenská zdravotnícka univerzita v Bratislave [online]. Bratislava, 2017 [cit. 2017-05-23]. Dostupné z: http://www.szu.sk/index.php?id=105&menu=20&kgid=153&idpart=4&iddp=2

[11] Radon and health. World Health Organisation [online]. World Health Organisation, 2017 [cit. 2017-05-23]. Dostupné z: http://www.who.int/mediacentre/factsheets/fs291/en/

[12] M.VIČANOVÁ, M.ĎURČÍK a D.NIKODEMOVÁ. Sledovanie výskytu radónu v podzemných pracovných priestoroch [online]. 21-22.10. Spišská Nová Ves, 1997

[cit. 2017-05-23]. Dostupné z: http://www.iaea.org/inis/collection/NCLCollectionStore/_Public/29/061/29061248.pdf

[13] Miscellaneous Detector Types. KNOLL, Glenn F. Radiation Detection and Measurement. 4th ed. Wiley, 2010, s. 736. ISBN 978-0470131480.

[14] ŠIKUDOVÁ, E., Z. ČERNEKOVÁ, W. BENEŠOVÁ, Z. HALADOVÁ a J. KUČEROVÁ. Počítačové videnie: Detekcia a rozpoznávanie objektov [online]. Praha: Wikina,Praha, 2013 [cit. 2016-11-23]. Dostupné z: http://vgg.fiit.stuba.sk/kniha/

[15] Dilation. *Image processing learning resources* [online]. Scotland: The university of Edinburgh, 2003 [cit. 2016-11-24]. Dostupné z: http://homepages.inf.ed.ac.uk/rbf/HIPR2/dilate.htm

[16] Erosion. *Image processing learning resources* [online]. Scotland: The university of Edinburgh, 2003 [cit. 2016-11-24]. Dostupné z: http://homepages.inf.ed.ac.uk/rbf/HIPR2/erode.htm

[17] Open. *Image processing learning resources* [online]. Scotland: The university of Edinburgh, 2003 [cit. 2016-11-24]. Dostupné z: http://homepages.inf.ed.ac.uk/rbf/HIPR2/open.htm

[18] Close. *Image processing learning resources* [online]. Scotland: The university of Edinburgh, 2003 [cit. 2016-11-24]. Dostupné z: http://homepages.inf.ed.ac.uk/rbf/HIPR2/close.htm

[19] Automated Upright Microscope System for Life Science Research Leica DM5500 B. *Leica microsystems* [online]. Leica Microsystems, 2016 [cit. 2016-12-11]. Dostupné z: http://www.leica-microsystems.com/products/light-microscopes/life-science-research/upright-microscopes/details/product/leica-dm5500-b/

[20] OASIS Interface for Leica Microsystems SmartMove Controller. Leica microsystems [online]. Leica Microsystems, 2016 [cit. 2016-11-24]. Dostupné z: http://objectiveimaging.com/OI_SmartMove.htm

[21] 6M Method for Cause and Effect Analysis. *Edraw Visualization Solutions* [online]. Jervois Street, Sheung Wan, HongKong: EdrawSoft, 2016 [cit. 2016-12-11]. Dostupné z: https://www.edrawsoft.com/6m-method.php

[22] KARPIŃSKA, Maria, Zenon MNICH a Jacek KAPAŁA. Seasonal changes in radon concentrations in buildings in the region of northeastern Poland. In: *Journal of Environmental Radioactivity* [online]. Volume 77, Issue 2. Elsevier, 2004, s. 101-109 [cit. 2016-12-11]. ISSN 0265-931X. Dostupné z: http://dx.doi.org/10.1016/S0265-931X(04)00226-7

[23] YUEN, H. K. a J. PRINCEN., J. Illingworth a J. Kittler. A comparative study of Hough transform methods for circle finding. In: Proc. 5th Alvey Vision Conf., Reading (31 Aug). Guildford: Department of Electronics and Electrical Engineering, 1989, s. 169-174.

[24] Hough Transform. In: *Wikipedia: the free encyclopedia* [online]. San Francisco (CA): Wikimedia Foundation, 2016 [cit. 2016-12-11]. Dostupné z: https://en.wikipedia.org/wiki/Hough_transform

[25] Hough Circle Transform. OpenCV 3.0.0-dev documentation [online]. opencv dev team, 2014 [cit. 2016-12-11]. Dostupné z: http://docs.opencv.org/3.0-beta/doc/py_tutorials/py_imgproc/py_houghcircles/py_houghcircles.html

[26] CANNY, John. A Computational Approach to Edge Detection. In: IEEE Trans. on Pattern Analysis and Machine Intelligence. 8(6). IEEE Publishing, 1986, s. 679-698.

[27] Feature Detection: HoughCircles. OpenCV 3.0.0-dev documentation [online]. opencv dev team, 2014 [cit. 2016-12-11]. Dostupné z: http://docs.opencv.org/3.0-beta/modules/imgproc/doc/feature_detection.html

[28] Open Source Computer Vision Library [online]. OpenCV Developers Team, 2016 [cit. 2016-12-13]. Dostupné z: http://opencv.org/

[29] Java Platform, Standard Edition 8 Names and Versions. Oracle [online]. *Oracle, 2017 [cit. 2017-05-23]. Dostupné z: http://www.oracle.com/technetwork/java/javase/jdk8-naming-2157130.html

[30] Neon. *Eclipse* [online]. The Eclipse Foundation, 2017 [cit. 2017-05-23]. Dostupné z: http://www.eclipse.org/neon/

[31] POI-HSSF and POI-XSSF - Java API To Access Microsoft Excel Format Files. Apache POI - the Java API for Microsoft Documents [online]. The Apache Software Foundation, 2017 [cit. 2017-05-23]. Dostupné z: https://poi.apache.org/spreadsheet/index.html

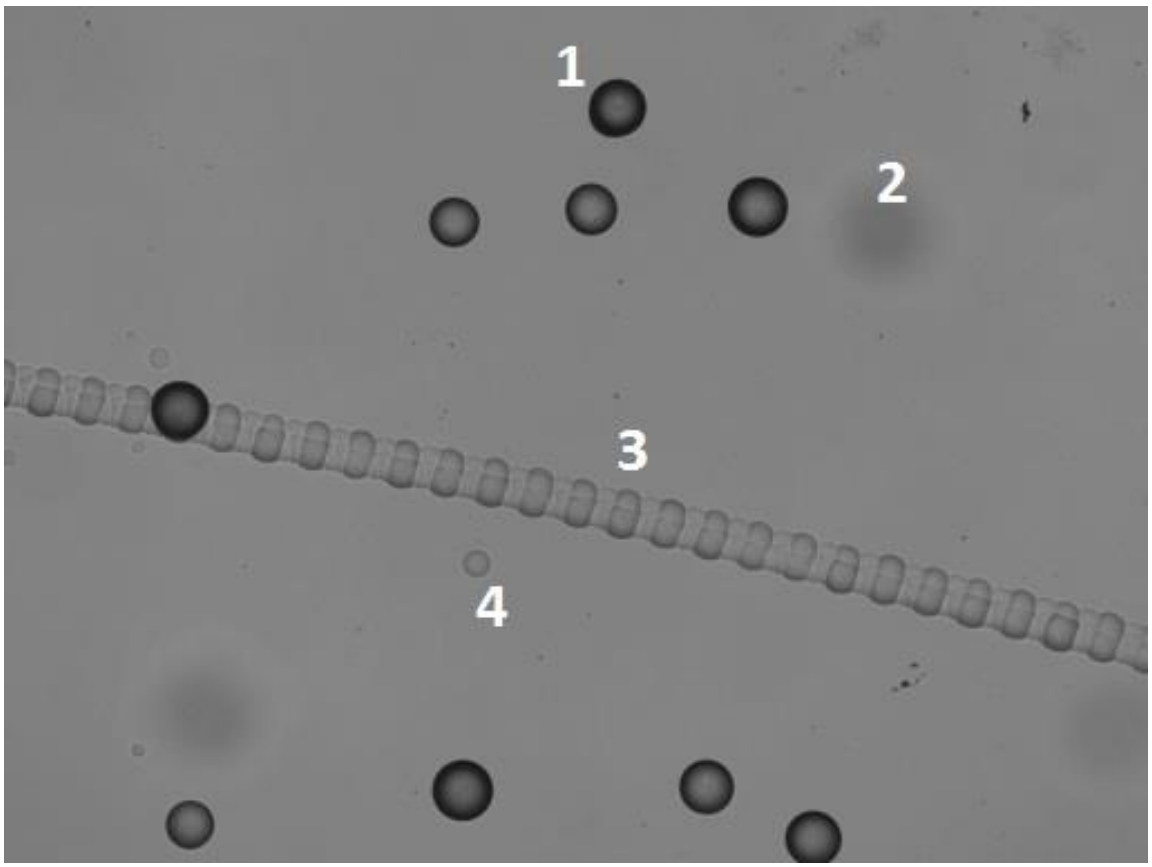
[32] Java DOM Parser - Create XML Document. Tutorials Point [online]. Madhapur, Hyderabad, 2017 [cit. 2017-05-23]. Dostupné z: https://www.tutorialspoint.com/java_xml/java_dom_create_document.htm

[33] European Platform for Occupational Radiation Exposure. Esorex platform [online]. EU: Esorex platform, 2017 [cit. 2017-05-23]. Dostupné z: https://esorex-platform.org/

Zoznam príloh

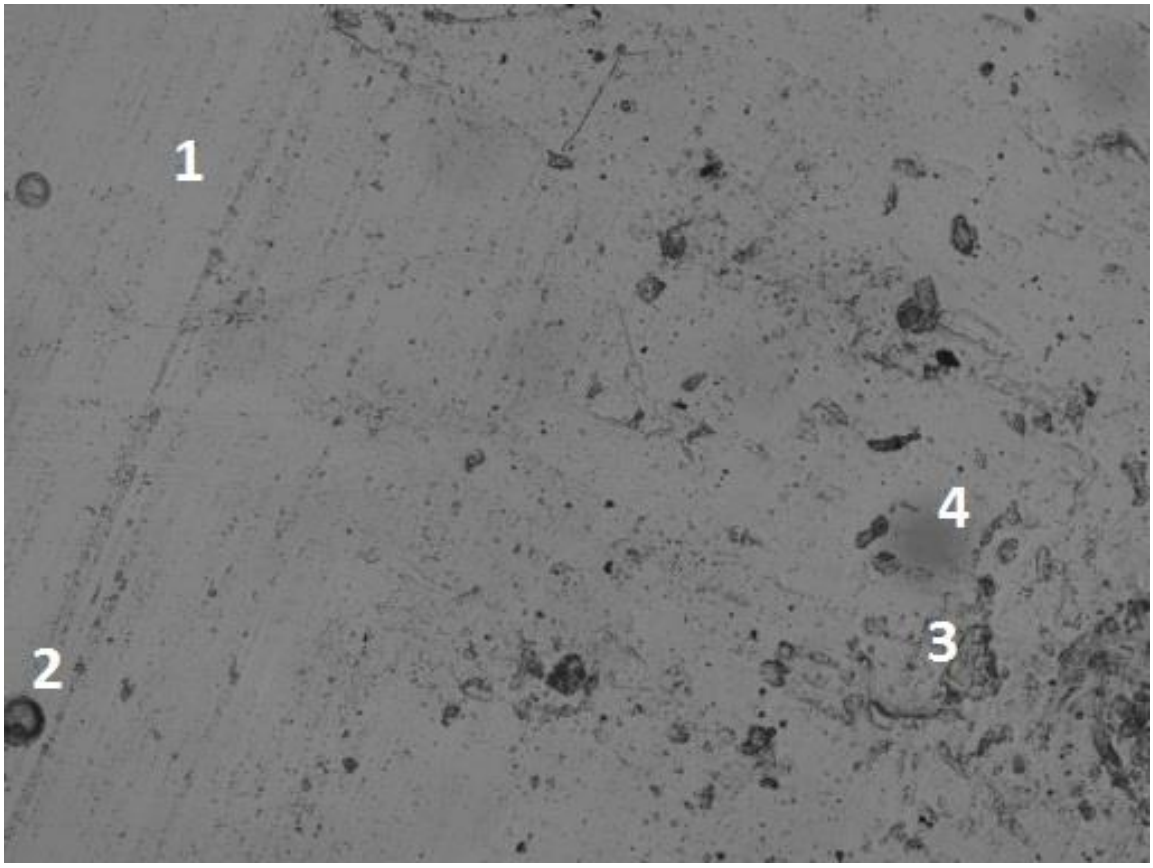
Príloha A. Príklady snímok segmentov exponovaného a vyleptaného dozimetra CR-39

A.1 Snímka segmentu s menej ožiareného dozimetra



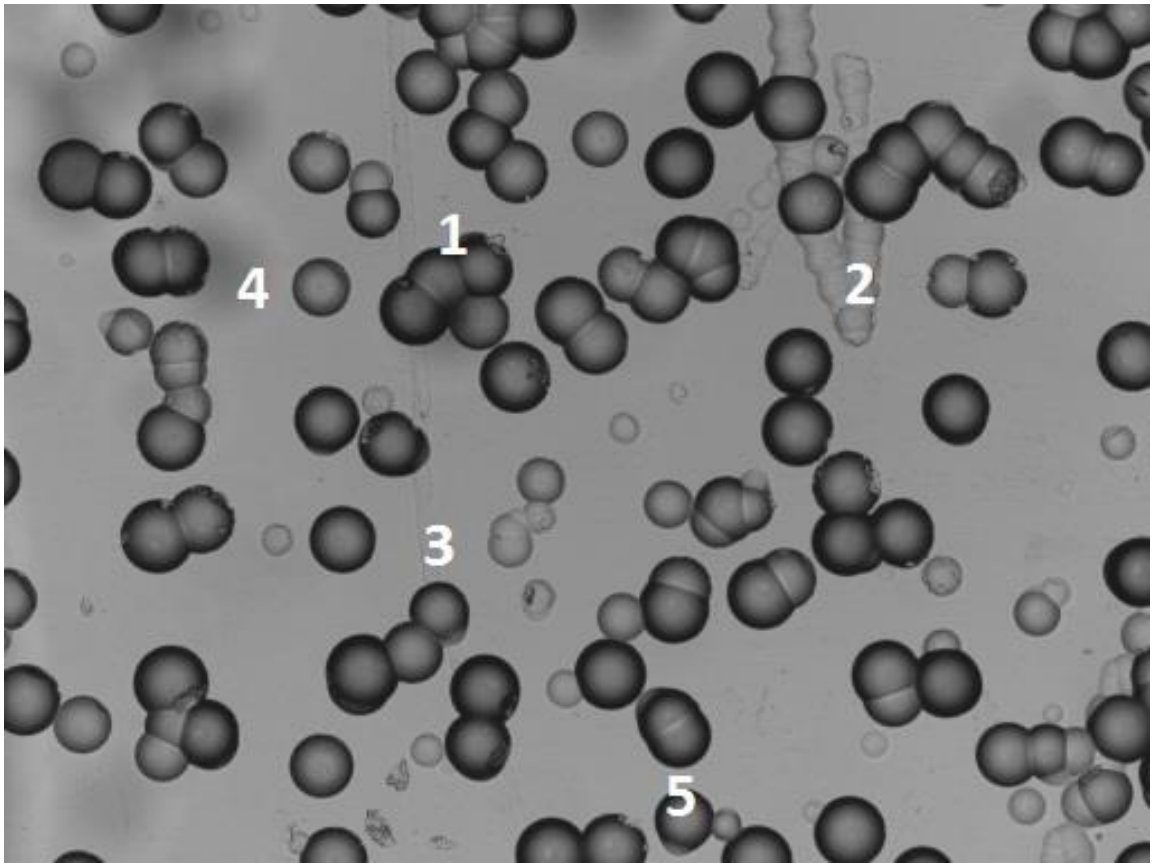
Na obrázku vidieť sivú plochu, čo je intenzita, ktorú vytvára materiál dozimetra, ktorý nieje úplne priesvitný. Snímka je zaostrená na stopy na povrchu. (1) - jasná veľká stopa vyleptaná na povrchu. (2) - stopa hlbšie v materiáli, ktorá nieje zaostrená. (3) – Reťazová stopa, ktorá obsahuje 2 rady bubliniek, bola spôsobená jednou časticou letiacou rovnobežne so snímanou štvorcovou plochou. (4) – malá, menej kontrastná stopa.

A.2 Snímka segmentu s množstvom škrabancov a defektov v obraze



Na obrázku vidieť: (1) - škrabance spôsobené pri procese výroby dozimetra a pri procese merania, (2) – stopa prekrytá menšou stopou, (3) – množstvo neforemných defektov, kt. sa nepodobajú na kruhový tvar, (4) – rozmazaná stopa na pozadí.

A.3 Snímka segmentu s množstvom zásahov alfa častíc



Na obrázku vidieť: (1) – väčšie množstvo prekrývajúcich sa zásahov, (2) – reťazová stopa s krivou dráhou spôsobená jednou časticou, možno dvoma časticami, (3) – reťazová stopa s menšími bublinami hlbšie v materiáli, (4) – rozostrená stopa hlbšie v materiáli, (5) – stopa, ktorá „má chvost“, resp. má pridruženú malú bublinu, ktorá sa takmer prekrýva s veľkou bublinou. Ide v podstate o reťazovú stopu, ktorá prechádza kolmo na sklo dozimetra, takže sa javí ako jedna bublina.

Príloha B. Zdrojový kód použitý pre ladenie parametrov Houghovej transformácie

Zdrojový kód je pre 32 bitový interpret python verzie 2.7.12 a používa knižnicu OpenCV2 vo verzii 3.0.0 a numpy vo verzii 1.11.1. Softvér bol testovaný na platforme s OS Windows 10 Home 64 bit.

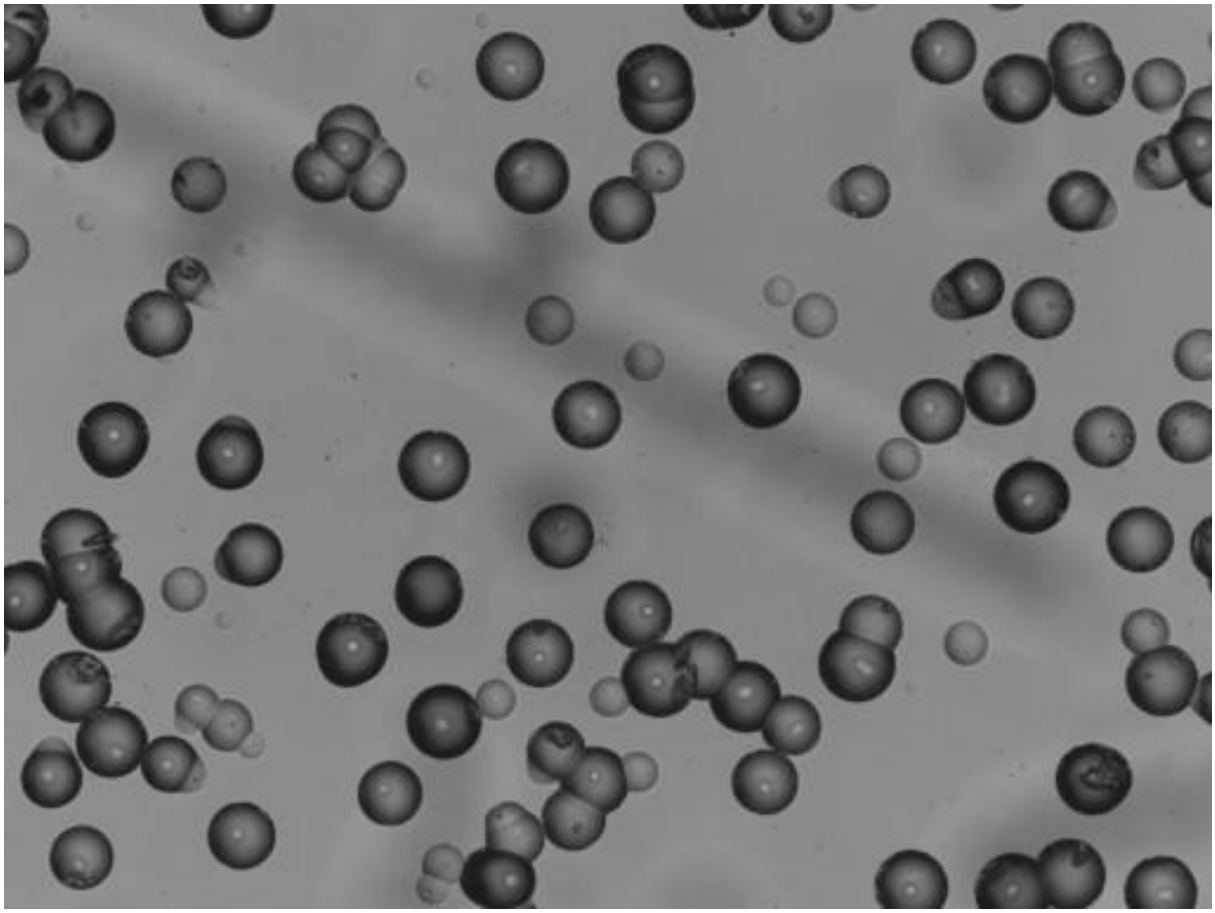
```
import cv2
import numpy as np
import sys
import os

outdir = os.path.join('.','testsout')
ff = 'IMG7.JPG'


def run(ifname,blur=5,mindist=20, param1=50, param2=30, minRadius=0,
maxRadius=100, dp=1):
    global outdir
    img = cv2.imread(ifname,0)
    if img is None:
        print "load image failed"
        sys.exit()

    if blur!=None:
        img = cv2.medianBlur(img,blur)

    cimg = cv2.cvtColor(img,cv2.COLOR_GRAY2BGR)

    circles = cv2.HoughCircles(img,cv2.HOUGH_GRADIENT,dp,mindist,
param1=param1,param2=param2,minRadius=minRadius,maxRadius=maxRadius)

    circles = np.uint16(np.around(circles))
    for i in circles[0,:]:
        # draw the outer circle
        cv2.circle(cimg,(i[0],i[1]),i[2],(0,255,0),2)
        # draw the center of the circle
        cv2.circle(cimg,(i[0],i[1]),2,(0,0,255),3)

    txt="min dist:"+str(mindist)
    cv2.putText(cimg, txt, (10,80), cv2.FONT_HERSHEY_TRIPLEX,
2,(255,255,255))
    cv2.putText(cimg, txt, (10+1,80), cv2.FONT_HERSHEY_TRIPLEX,
2,(255,255,255))

    txt="dp:"+str(dp)
    cv2.putText(cimg, txt, (10,80+80), cv2.FONT_HERSHEY_TRIPLEX,
2,(255,255,255))
    cv2.putText(cimg, txt, (10+1,80+80), cv2.FONT_HERSHEY_TRIPLEX,
2,(255,255,255))

    txt="param1:"+str(param1)
    cv2.putText(cimg, txt, (600+10,80), cv2.FONT_HERSHEY_TRIPLEX,
2,(255,255,255))
```

```
    cv2.putText(cimg, txt, (600+10+1,80), cv2.FONT_HERSHEY_TRIPLEX,
2,(255,255,255))

    txt="param2:"+str(param2)
    cv2.putText(cimg, txt, (600+10,80+80), cv2.FONT_HERSHEY_TRIPLEX,
2,(255,255,255))
    cv2.putText(cimg, txt, (600+10+1,80+80), cv2.FONT_HERSHEY_TRIPLEX,
2,(255,255,255))

    txt="minRadius:"+str(minRadius)
    cv2.putText(cimg, txt, (1200+10,80), cv2.FONT_HERSHEY_TRIPLEX,
2,(255,255,255))
    cv2.putText(cimg, txt, (1200+10+1,80), cv2.FONT_HERSHEY_TRIPLEX,
2,(255,255,255))

    txt="maxRadius:"+str(maxRadius)
    cv2.putText(cimg, txt, (1200+10,80+80), cv2.FONT_HERSHEY_TRIPLEX,
2,(255,255,255))
    cv2.putText(cimg, txt, (1200+10+1,80+80),
cv2.FONT_HERSHEY_TRIPLEX, 2,(255,255,255))


    ofname = "aa_b" + str(blur) +
"md"+str(mindist)+"p"+str(param1)+"q"+str(param2)+"minr"+str(minRadius
)+"maxr"+str(maxRadius)+"dp"+str(dp)
    ofpath = os.path.join(outdir,ofname+".jpg")
    cv2.imwrite(ofpath, cimg)
    print "processed " + ofpath

    cv2.line(cimg, (10, 20), (10+mindist,20), (255,255,255),
thickness=5, lineType=8, shift=0)

    #cv2.namedWindow("detected circles", cv2.WINDOW_NORMAL)
    #cv2.imshow('detected circles',cimg)
    #cv2.waitKey(0)
    #cv2.destroyAllWindows()

def test1(blur):
    for i in range(1, 5):
        run(ff,blur=blur,mindist=20, param1=50, param2=i, minRadius=0,
maxRadius=100, dp=1)

    for j in range(1, 10):
        i=5*j
        run(ff,blur=blur,mindist=20, param1=50, param2=i, minRadius=0,
maxRadius=100, dp=1)

def test2(blur):
    for j in range(2, 10):
        i=5*j
        run(ff,blur=blur,mindist=20, param1=i, param2=30, minRadius=0,
maxRadius=100, dp=1)
        run(ff,blur=blur,mindist=20, param1=150, param2=30,
minRadius=0, maxRadius=100, dp=1)
        run(ff,blur=blur,mindist=20, param1=200, param2=30,
minRadius=0, maxRadius=100, dp=1)
        run(ff,blur=blur,mindist=20, param1=250, param2=30,
minRadius=0, maxRadius=100, dp=1)
```

```
        run(ff,blur=blur,mindist=20, param1=300, param2=30,
minRadius=0, maxRadius=100, dp=1)
        run(ff,blur=blur,mindist=20, param1=350, param2=30,
minRadius=0, maxRadius=100, dp=1)
        run(ff,blur=blur,mindist=20, param1=400, param2=30,
minRadius=0, maxRadius=100, dp=1)
        #run(ff,blur=blur,mindist=20, param1=450, param2=30,
minRadius=0, maxRadius=100, dp=1)
        #run(ff,blur=blur,mindist=20, param1=500, param2=30,
minRadius=0, maxRadius=100, dp=1)
        #run(ff,blur=blur,mindist=20, param1=550, param2=30,
minRadius=0, maxRadius=100, dp=1)

      def test3(blur):
        run(ff,blur=blur,mindist=20, param1=50, param2=30,
minRadius=0, maxRadius=100, dp=1)
        run(ff,blur=blur,mindist=20, param1=50, param2=30,
minRadius=0, maxRadius=100, dp=2)
        run(ff,blur=blur,mindist=20, param1=50, param2=30,
minRadius=0, maxRadius=100, dp=3)
        run(ff,blur=blur,mindist=20, param1=50, param2=30,
minRadius=0, maxRadius=100, dp=4)

      def test4(blur):
        for i in range(1,20):
              dp=5*i
              run(ff,blur=blur,mindist=20, param1=50, param2=30,
minRadius=0, maxRadius=100, dp=dp)

      def test5(blur):
          for i in
[1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,25,30,35,40,45,50]
:
              run(ff,blur=blur,mindist=i, param1=50, param2=30,
minRadius=0, maxRadius=100, dp=1)

      def test6(blur):
          for i in
[1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,25,30,35,40,45,50]
:
              run(ff,blur=blur,mindist=20, param1=50, param2=30,
minRadius=i, maxRadius=100, dp=1)



      test1(9)
      test2(9)
      test3(9)
      test4(9)
      test5(9)
      test6(9)
```

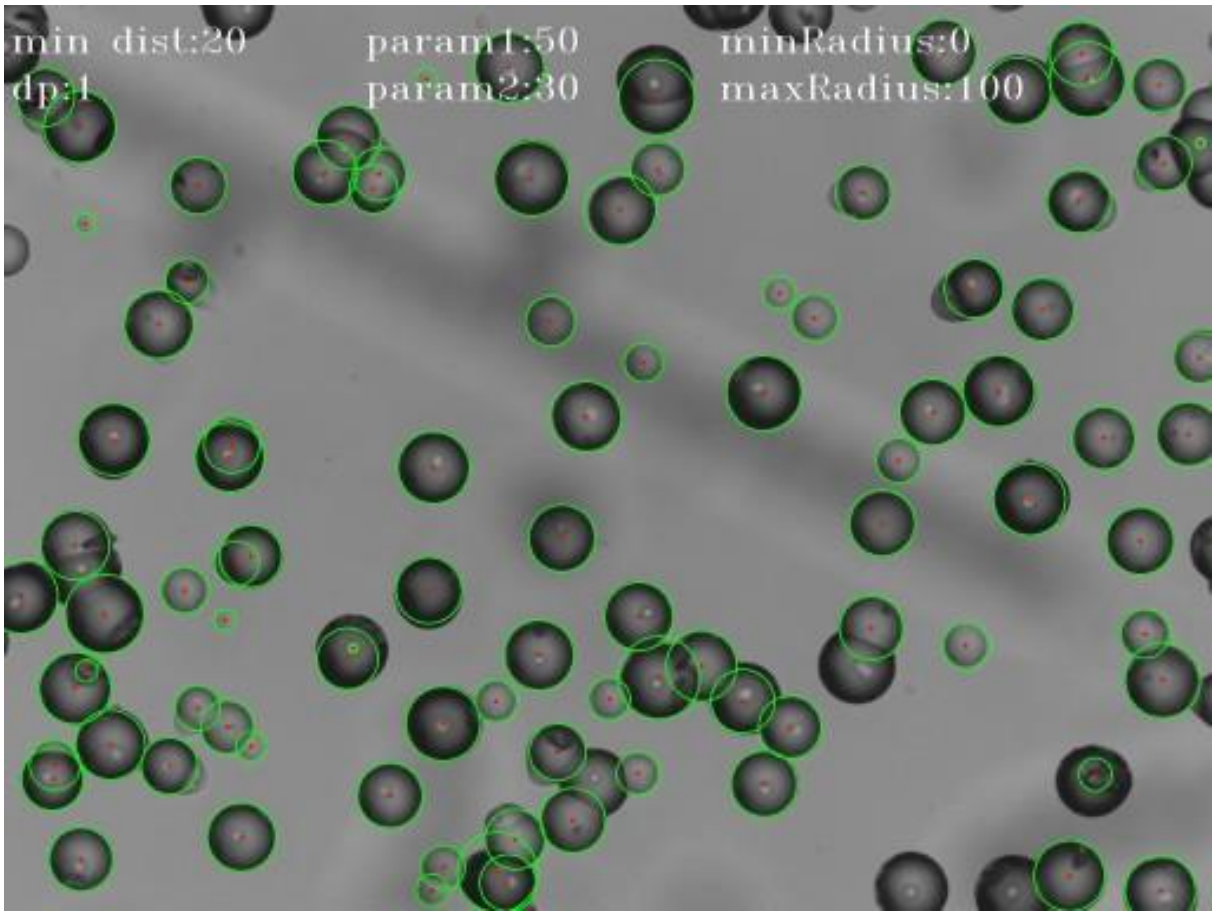
Príloha C. Testovacia snímka a výsledky testovania

C.1 Testovacia snímka



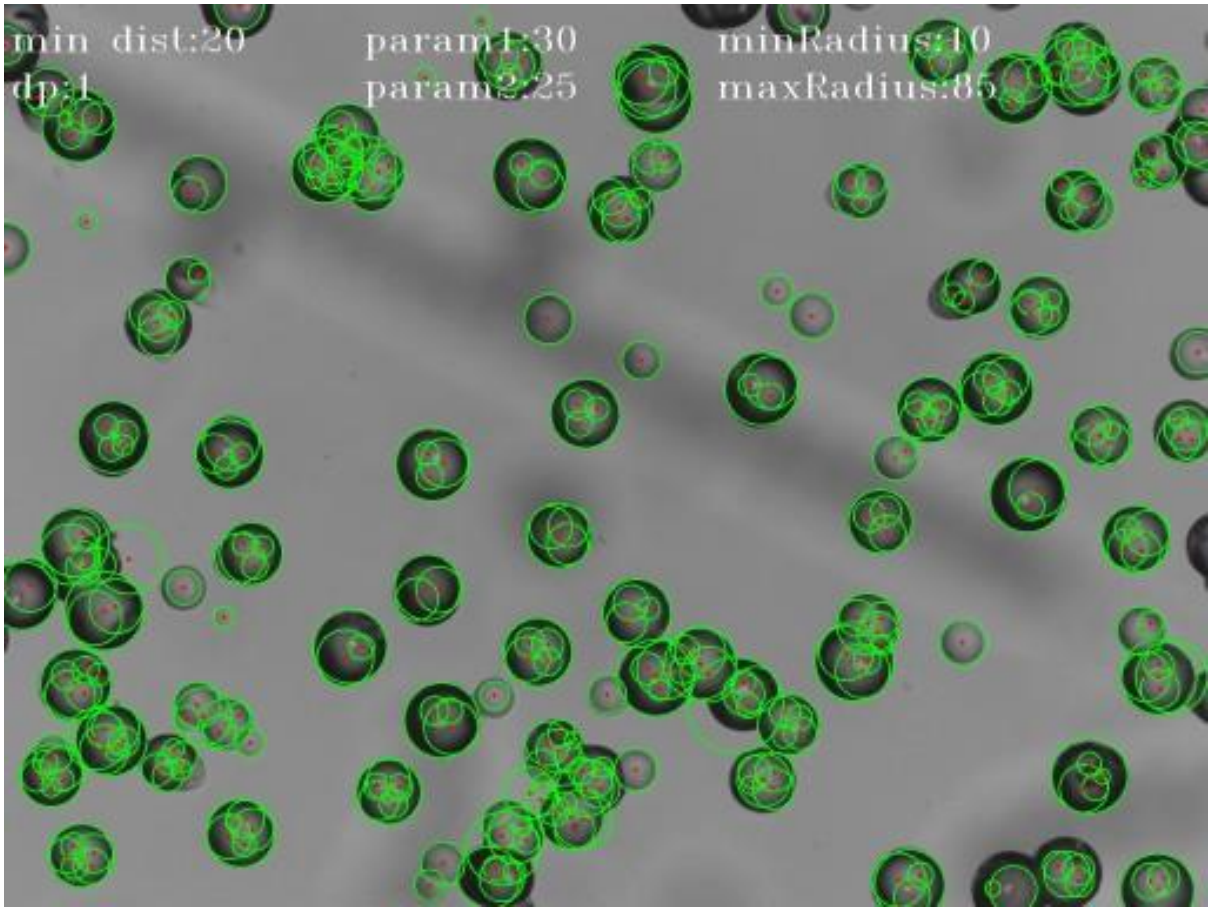
Snímka obsahuje väčší počet prekrývajúcich sa zásahov, taktiež obsahuje stopy s chvostom a na pozadí sa nachádzajú rozmazané stopy hlbšie v materiáli. Taktiež sú prítomné malé stopy, ktoré môže byť problém detegovať. Na snímke boli detegované kruhy pomocou Houghovej transformácie skriptom v jazyku python v prílohe B.

C.2 Počiatočné nastavenie transformácie



Detekcia bola robená s prvotnými skusmo ladenými parametrami, od ktorých bolo odvodené rozmietanie ostatných parametrov. Neboli detegované stopy ležiace zväčša pri okraji, pri spodnom okraji vidieť stopu, ktorej priemer bol nadhodnotený. Viacerým stopám nebol detegovaný chvost, vďaka čomu detekciu netreba filtrovať na tieto prípady.

C.3 Detekcia s nájdenými parametrami



Detekcia s parametrami, ktoré nestrácajú detekciu väčšiny stôp, ktoré ležia na okraji snímky. Je produkovaných vela falošných detekcií, z ktorých by sa väčšina mala dať odfiltrovať vhodným algoritmom.

Príloha D. Tabuľka počtu stôp z pôvodnej a novej metódy

V stĺpci jaskyňa je názov adresára s dozimetrami z danej jaskyne z adresára dozimetre na DVD. Stĺpec Dozimeter je sériové číslo priradené dozimetru na jeho presnú identifikáciu. Stĺpec “Počet stôp

Jaskyňa	Dozimeter	Počet stôp	
		pôvodná	nová
belianska	AE1	879	641
belianska	AE17	1077	895
belianska	AE2	1719	1942
belianska	AE27	984	723
belianska	AE3	841	858
belianska	AE4	1100	1224
belianska	AE46	1289	753
belianska	AE47	1385	611
belianska	AE48	1707	1051
belianska	AE7	1264	830
belianska	AE8	1213	519
belianska	AF01	1416	826
belianska	AF02	793	756
belianska	AF03	1706	959
belianska	AF04	1253	1386
belianska	AF05	1381	2489
belianska	AF06	1176	1093
belianska	AF41	1332	746
belianska	AF45	1099	649
bystrianska	AE21	3187	2911
bystrianska	AE22	4020	8611
bystrianska	AE23	2652	1823
bystrianska	AE24	2579	1372
bystrianska	AE26	3130	2852
bystrianska	AE38	1655	1186
bystrianska	AF43	2817	2036
bystrianska	AF53	3097	2296
bystrianska	AF54	2673	2609
bystrianska	AF56	3251	3076
bystrianska	AF65	2175	7311
demenov	AH1	1432	815
demenov	AH11	899	779
demenov	AH12	1505	1653
demenov	AH14	1443	1424

demenov	AH15	1307	1117
demenov	AH16	1603	1046
demenov	AH17	2567	893
demenov	AH19	1669	1093
demenov	AH2	1766	910
demenov	AH21	1497	1013
demenov	AH23	1322	1215
demenov	AH24	2195	1416
demenov	AH27	1397	786
demenov	AH28	1717	1329
demenov	AH30	1300	1364
demenov	AH32	1603	991
demenov	AH36	1786	1136
demenov	AH38	1234	823
demenov	AH40	1933	1261
demenov	AH43	1975	1260
demenov	AH7	1564	877
demenov	AH8	1372	1302
domica	AI34	1282	1542
domica	AI53	1590	1988
domica	AI56	2016	2422
domica	AI61	1654	2004
domica	AI62	2032	2157
domica	AI65	1476	2098
domica	AI66	1800	2124
domica	AI67	1471	1880
domica	AI68	1543	2084
driny	AI33	3382	5619
driny	AI35	3348	7314
driny	AI36	3159	4896
driny	AI37	3065	5075
driny	AI40	3395	6421
driny	AI41	3538	5733
driny	AI45	2928	5350
driny	AI46	3671	5279
driny	AI48	3292	6158
gombasek	AI1	2404	1014
gombasek	AI15	1092	698
gombasek	AI17	2494	1232
gombasek	AI19	3078	1616
gombasek	AI2	1205	2215
gombasek	AI3	2000	928
gombasek	AI31	3397	1512

harmanec	AI10	1719	2098
harmanec	AI11	2155	3407
harmanec	AI12	1623	2156
harmanec	AI16	1965	2706
harmanec	AI39	1972	2340
harmanec	AI42	2430	6274
harmanec	AI43	2039	2752
harmanec	AI47	2028	2416
harmanec	AI7	1668	2232
jasov	AH49	2955	3142
jasov	AH52	2643	2674
jasov	AH54	2793	2741
jasov	AH67	2982	5179
jasov	AH70	2417	2719
jasov	AH71	3153	3056
jasov	AH72	3088	5107
ochtinska	AE70	2818	9449
ochtinska	AF08	2651	7332
ochtinska	AF1	2605	8914
ochtinska	AF41	2755	7874
ochtinska	AF45	2355	9487
ochtinska	AF46	2807	10338
ochtinska	AF47	2602	7248
ochtinska	AH61	2562	8309
ochtinska	AH62	2678	7429
ochtinska	AH63	2593	7733
ochtinska	AH64	2628	6993
ochtinska	AH66	2843	10511
ochtinska	AH68	2453	8144
ochtinska	AH69	760	10169
ochtinska	AI42	1962	2719
vazecka	1_AH55	2003	1901
vazecka	2_AH48	2879	3237
vazecka	3_AH50	2799	4962
vazecka	4_AH56	2474	5604
vazecka	5_AH58	2727	4003
vazecka	6_AH47	2667	5998
vazecka	7_AH46	3262	4100
vazecka	P_AH57	2553	1799

Príloha E. Obsah DVD

Na dvd sa nachádzajú tieto adresáre:

dozimetre	Adresár obsahuje pôvodné snímky zo 116 vyvolanách dozimetrov z 10 slovenských jaskýň za 3. štvrťrok 2015. Ďalej sa v adresári nachádzajú skripty v jazyku python, ktoré vytvárajú zoznam jaskýň a adresárov so snímkami.
kod	V tomto adresári sa nachádza kód vytvorenej aplikácie na spracovanie dozimetrov. Použité prostredie bolo Java Eclipse Neon s pluginom Window Builder.
ladenieparam	Tu sa nachádzajú skripty v jazyku python, pomocou ktorých boli ladené parametre Houghovej transformácie.
program	V tomto adresári sa nachádza spustiteľná verzia programu s priloženým java runtime
statistika	Tu sú štatistické excelové workbooky obsahujúce dáta na porovnanie detekčných metód.
tlac	Tu sú súbory určené na tlač diplomovej práce.